



TUGAS AKHIR - TF 141581

PERANCANGAN SISTEM *ACTIVE FAULT TOLERANT CONTROL* (AFTC) UNTUK PENGENDALIAN POSISI DENGAN KONTROL *CASCADE* PADA SISTEM SERVO MODULAR MS150 DC

Tulus Indra Hermawan
NRP. 02311340000018

Dosen Pembimbing :
Dr. Katherin Indriawati, S.T., M.T.

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2018

“Halaman ini memang dikosongkan”



FINAL PROJECT - TF 141581

***DESIGN ACTIVE FAULT TOLERANT
CONTROL SYSTEM (AFTC) FOR POSITION
CONTROL WITH CASCADE CONTROL OF
MODULAR SERVO SYSTEM MS150 DC***

Tulus Indra Hermawan
NRP. 02311340000018

Supervisors :
Dr. Katherin Indriawati, S.T., M.T.

ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2018

“Halaman ini memang dikosongkan”

PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini

Nama : Tulus Indra Hermawan
NRP : 02311340000018
Departemen/ Prodi : Teknik Fisika/ S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "Perancangan sistem *Active Fault Tolerant Control (AFTC)* untuk pengendalian posisi dengan kontrol *cascade* pada sistem servo modular MS150 DC " adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada tugas akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 10 Januari 2018
Yang membuat pernyataan,



Tulus Indra Hermawan
NRP. 02311340000018

“Halaman ini memang dikosongkan”

LEMBAR PENGESAHAN

PERANCANGAN SISTEM *ACTIVE FAULT TOLERANT CONTROL (AFTC)* UNTUK PENGENDALIAN POSISI DENGAN KONTROL *CASCADE* PADA SISTEM SERVO MODULAR MS150 DC

Oleh:

Tulus Indra Hermawan

NRP. 02311340000018

Surabaya, 10 Januari 2018

Menyetujui,
Dosen Pembimbing



Dr. Katherin Indriawati, S.T., M.T.

NIPN. 19760523 200012 2 001

Mengetahui,

Kepala Departemen

Teknik Fisika FTI-ITS



Agus Muhammad Hatta, S.T., M.Si., Ph.D.

NIPN. 197809022003121002

“Halaman ini memang dikosongkan”

PERANCANGAN SISTEM *ACTIVE FAULT TOLERANT CONTROL (AFTC)* UNTUK PENGENDALIAN POSISI DENGAN KONTROL *CASCADE* PADA SISTEM SERVO MODULAR MS150 DC

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada


Bidang Studi Rekayasa Instrumentasi
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh:


Tulus Indra Hermawan
NRP. 02311340000018

Disetujui oleh Tim Penguji Tugas Akhir:

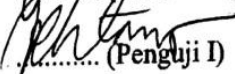
1. Dr. Katherin Indriawati, S.T., M.T.

.....(Pembimbing)

2. Dr. Imam Abadi, S.T., M.T.

.....(Ketua Penguji)

4. Dr. Ridho Hantoro, S.T., M.T.

.....(Penguji I)

SURABAYA
Januari, 2018

“Halaman ini memang dikosongkan”

PERANCANGAN SISTEM *ACTIVE FAULT TOLERANT CONTROL (AFTC)* UNTUK PENGENDALIAN POSISI DENGAN KONTROL *CASCADE* PADA SISTEM SERVO MODULAR MS150 DC

Nama : Tulus Indra Hermawan
NRP : 02311340000018
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Dr. Katherin Indriawati, S.T., M.T.

Abstrak

Kesalahan pada komponen seperti aktuator dan sensor dapat mengakibatkan kegagalan sistem salah satunya pada pengendalian posisi motor DC. Pada tugas akhir ini dilakukan perancangan sistem *Active Fault Tolerant Control (AFTC)* yang mampu bekerja ketika terdapat kesalahan pada komponen seperti aktuator dan sensor pada pengendalian posisi sistem servo modular MS150 DC sehingga performansi sistem tetap terjaga. Langkah pertama yang dilakukan adalah membuat pemodelan servo modular MS150 DC. Langkah kedua adalah merancang sistem kontrol *cascade* PI-P. Langkah ketiga adalah merancang sistem AFTC secara simulasi dan langkah terakhir adalah merancang sistem AFTC untuk aplikasi *real time*. Sistem tanpa AFTC pada kesalahan sensor secara *real time* mengalami *error steady state* sebesar 11% dan 91% sedangkan sistem dengan AFTC tidak mengalami *error steady state*. Sistem tanpa AFTC pada kesalahan aktuator secara *real time* mengalami *overshoot* sebesar 12,7% dan 28,6%, sedangkan sistem dengan AFTC mengalami *overshoot* lebih kecil sebesar 1,59% dan 11,15%. Hal tersebut terjadi karena sistem AFTC bisa menolerir kesalahan dari sensor dan aktuator sehingga respon sistem menjadi lebih baik.

Kata Kunci: Sensor, Aktuator, *Cascade* PI-P, Sistem AFTC, Sistem MS150 DC.

“Halaman ini memang dikosongkan”

**DESIGN ACTIVE FAULT TOLERANT CONTROL SYSTEM
FOR POSITION CONTROL WITH CASCADE CONTROL
OF MODULAR SERVO SYSTEM MS150 DC**

Name : **Tulus Indra Hermawan**
NRP : **02311340000018**
Department : **Teknik Fisika FTI-ITS**
Supervisors : **Dr. Katherin Indriawati, S.T., M.T.**

Abstract

Errors in components in actuators and sensors can affect to system failure such as at position control of DC motor. In this final project, Active Fault Tolerant Control (AFTC) system design is able to work when there are errors in components such as actuators and sensors on position control of MS150 DC modular servo system so it can maintain the system performance. The first step is to make model of modular servo MS150 DC. The second step is to design a PI-P cascade control system. The third step is to design AFTC system simulation and final step is to design AFTC system for real time application. The system without AFTC on sensor error in real time generated error steady state 11% and 91% and the system with AFTC did not generate steady state error. System without AFTC on actuator error in real time generated overshoot 12,7% and 28,6%, and the system with AFTC has smaller overshoot equal to 1,59% and 11,15%. It happens because the AFTC system can tolerate errors from sensors and actuators so the system response can becomes better.

Keywords : Sensor, Actuator, Cascade PI-P, AFTC system, MS150 DC system

“Halaman ini memang dikosongkan”

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang senantiasa melimpahkan rahmat serta hidayah-Nya, serta shalawat serta salam kepada Nabi Muhammad SAW, hingga terselesaikannya tugas akhir beserta laporan tugas akhir yang berjudul

PERANCANGAN SISTEM *ACTIVE FAULT TOLERANT CONTROL* UNTUK PENGENDALIAN POSISI DENGAN KONTROL *CASCADE* PADA SISTEM SERVO MODULAR MS150 DC

Penulis telah banyak memperoleh bantuan dari berbagai pihak dalam penyelesaian tugas akhir dan laporan Tugas Akhir ini. Penulis mengucapkan terimakasih kepada :

1. Bapak Agus Muhamad Hatta, S.T., M.Si., Ph.D selaku Kepala Departemen Teknik Fisika yang telah memberikan petunjuk, ilmu, serta bimbingan selama menempuh pendidikan di Teknik Fisika.
2. Ibu Dr. Katherin Indriawati, S.T., M.T. selaku dosen pembimbing yang telah dengan sabar memberikan petunjuk, ilmu, serta bimbingan yang sangat bermanfaat.
3. Bapak Totok Ruki Biyanto, S.T., M.T., Ph.D. selaku Kepala Laboratorium Rekayasa Instrumensi yang telah memberikan ilmu, petunjuk, nasihat, serta kemudahan perizinan.
4. Bapak Ir.Zulkifli selaku dosen wali yang telah membimbing penulis selama perkuliahan.
5. Kedua orang tua Bapak Heri Widodo dan ibu Ainin Kholianah terimakasih atas segala cinta, kasih sayang, doa, perhatian, serta dukungan moril dan materiil yang telah diberikan.
6. Seluruh teman – teman Departemen Teknik Fisika angkatan 2013, terima kasih untuk dukungannya.
7. Seluruh dosen, karyawan dan civitas akademik Teknik Fisika, terimakasih atas segala bantuan dan kerjasamanya.
8. Semua pihak yang tidak dapat disebutkan satu persatu, terimakasih atas bantuannya.

Penulis sadar bahwa penulisan laporan tugas akhir ini tidak sempurna, namun semoga laporan ini dapat memberikan kontribusi yang berarti dan menambah wawasan yang bermanfaat bagi pembaca, keluarga besar Teknik Fisika khususnya, dan civitas akademik ITS pada umumnya. Semoga laporan tugas akhir ini dapat bermanfaat sebagai referensi pengerjaan laporan tugas akhir bagi mahasiswa yang lain.

Surabaya, 10 Januari 2018

Penulis

DAFTAR ISI

	Halaman
Halaman Judul.....	i
<i>Title Page</i>	iii
Lembar Pengesahan I	xvii
Lembar Pengesahan II	ix
Abstrak	xi
<i>Abstract</i>	xviii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR NOTASI	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Sistematika Laporan.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 Motor DC	5
2.2 Sistem Servo Modular MS150 DC	6
2.3 Model Matematik Motor DC	7
2.4 Sistem Pengendalian <i>Cascade</i>	11
2.5 Sistem Kendali PI-P	12
2.6 Sistem <i>Active Fault Tolerant Control (AFTC)</i>	14
2.7 <i>Observer</i>	16
BAB III METODOLOGI PENELITIAN	19
3.1 Studi Literatur	21
3.2 Set Eksperimen Sistem Servo Modular MS150 DC	21
3.3 Pemodelan Servo Modular MS150 DC.....	22
3.4 Validasi Pemodelan.....	25
3.5 Perancangan Sistem Kendali <i>Cascade</i> PI-P	26
3.6 Perancangan Sistem <i>Active Fault Tolerant Control</i> (AFTC) Secara Simulasi	26

3.7	Perancangan Sistem <i>Active Fault Tolerant Control</i> (AFTC) Secara <i>Real Time</i>	30
BAB IV ANALISIS HASIL DAN PEMBAHASAN.....		33
4.1	Validasi Data	33
4.2	Sistem Kendali <i>Cascade</i> PI-P	34
4.3	Uji Performansi	36
BAB V PENUTUP		45
5.1	Kesimpulan	45
5.2	Saran.....	45
DAFTAR PUSTAKA.....		47
LAMPIRAN		49
BIODATA PENULIS		

DAFTAR GAMBAR

Gambar 2.1. Motor DC	6
Gambar 2.2. Sistem servo modular MS150 DC	6
Gambar 2.3. Rangkaian motor DC	8
Gambar 2.4. Skema pengendalian cascade	11
Gambar 2.5. Diagram blok pengendalian PI.....	12
Gambar 2.6. Struktur umum sistem AFTC.....	15
Gambar 2.7. <i>Observer</i>	17
Gambar 2.8. Kestabilan <i>pole</i> sistem diskrit	18
Gambar 3.1. Diagram alir penelitian tugas akhir.....	20
Gambar 3.2. Diagram set eksperimen sistem MS150 DC	21
Gambar 3.3. Set eksperimen <i>real plant</i> sistem MS150 DC	22
Gambar 3.4. Diagram pengendalian <i>cascade</i> P-PI	25
Gambar 4.1. Respon sistem validasi kecepatan motor DC.....	33
Gambar 4.2. Respon sistem validasi posisi motor DC.....	34
Gambar 4.3. Respon sistem perancangan sistem kendali PI-P secara simulasi.....	35
Gambar 4.4. Respon sistem penerapan sistem kendali PI-P secara <i>real time</i>	36
Gambar 4.5. Respon sistem AFTC dengan kesalahan bias 16% pada aktuator dan 8% pada sensor secara simulasi.....	37
Gambar 4.6. Respon sistem AFTC dengan kesalahan bias 63,7% pada aktuator dan 63,7% pada sensor secara simulasi.....	38
Gambar 4.7. Respon sistem dengan kesalahan bias pada aktuator sebesar 16% secara <i>real time</i>	39
Gambar 4.8. Respon sistem hasil AFTC dengan kesalahan bias pada aktuator sebesar 16% secara <i>real time</i>	40
Gambar 4.9. Respon sistem dengan kesalahan bias pada aktuator sebesar 63,7% secara <i>real time</i>	40
Gambar 4.10. Respon sistem hasil AFTC dengan kesalahan bias pada aktuator sebesar 63,7% secara <i>real time</i>	41
Gambar 4.11. Respon sistem dengan kesalahan bias pada sensor sebesar 8% secara <i>real time</i>	42

Gambar 4.12. Respon sistem dengan kesalahan bias pada sensor sebesar 63,7% secara *real time*42

DAFTAR TABEL

Tabel 4.1. Performansi sistem dengan kesalahan bias 16% pada aktuator dan 8% pada sensor secara simulasi.....	37
Tabel 4 2. Performansi sistem dengan kesalahan bias 63,7% pada aktuator dan 63,7% pada sensor secara simulasi.....	38
Tabel 4.3. Performansi sistem dengan kesalahan bias pada aktuator 16% dan 63,7% secara <i>real time</i>	41
Tabel 4.4. Performansi sistem dengan kesalahan bias pada sensor 8% dan 63,7% secara <i>real time</i>	43

“Halaman ini memang dikosongkan”

DAFTAR NOTASI

K_t	= Konstanta torsi motor
i	= Arus motor
T	= Torsi motor
b	= Koefisien gesekan viskos ekivalen dari motor (Nm/rad/s)
ω	= Kecepatan rotasi (rad/s)
J	= Momen inersia ekivalen dari motor (Kgm ²)
L	= induktansi kumparan jangkar (H)
R	= Tahanan kumparan jangkar (A)
e_a	= Tegangan kumparan jangkar (V)
e_b	= Gaya gerak listrik balik (V)
V	= Tegangan input (V)
K	= Konstanta motor
θ	= posisi motor
A	= Matriks keadaan
B	= Matriks masukan
C	= Matriks keluaran
D	= Matriks gangguan
K_p	= Konstanta proporsional
K_i	= Konstanta integral
T_s	= Waktu cuplik
τ_i	= Konstanta waktu integral
$x(k+1)$	= Turunan vector keadaan
$\hat{x}(k+1)$	= Turunan estimasi $x(k+1)$
K_e	= Gain Observer
$\hat{x}(k)$	= Estimasi vektor keadaan
$\hat{y}(k)$	= Estimasi vektor keluaran
I	= Matriks identitas
z	= <i>Pole</i>

“Halaman ini memang dikosongkan”

BAB I

PENDAHULUAN

1.1 Latar Belakang

Motor DC (*Direct Current*) merupakan perangkat elektromagnetis yang berfungsi mengubah energi listrik menjadi energi mekanik. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Motor arus searah tidak biasa digunakan di aplikasi industri karena semua sistem utilitas listrik menyediakan arus bolak-balik. Namun, untuk aplikasi khusus seperti di pabrik baja, tambang dan kereta listrik terkadang menguntungkan untuk mengubah arus bolak-balik menjadi arus searah dalam penggunaan motor DC. Alasannya adalah karakteristik torsi kecepatan motor dapat divariasikan dengan efisiensi yang tinggi (Wildi, 2002).

Salah satu subjek penelitian motor DC yang sering ditemui adalah pengendalian posisi yang diaplikasikan dalam berbagai bidang seperti transportasi, peralatan elektronik, industri dan robotika. Desain pengendalian posisi yang sering digunakan adalah *closed loop* yang merupakan pengendalian paling umum dan sederhana (Namazov, 2010).

Dalam pengendalian posisi motor DC satu loop, saat *gain* ditingkatkan maka input tegangan yang dibutuhkan untuk menghasilkan output akan berkurang. Namun hal tersebut berdampak dengan meningkatnya *overshoot* sehingga sistem menjadi tidak stabil. Untuk mengatasi hal tersebut, maka digunakan sistem pengendalian tertutup dengan kontrol *cascade* (Arvani, 2007).

Suatu sistem pengendalian memiliki kinerja yang buruk ketika terjadi kesalahan pada komponen, misalnya pada sensor dan aktuator. Untuk mengatasi kelemahan tersebut maka dikembangkan pendekatan baru untuk menoleransi kerusakan komponen sehingga performansi dan kestabilan sistem tetap terjaga serta dapat mencapai kondisi yang optimal. Sistem pengendalian yang memiliki kemampuan untuk mempertahankan

kestabilan dan performansi sistem pada saat terjadi kesalahan pada komponen sensor dan aktuator disebut *Fault Tolerant Control System* (FTCS). *Fault Tolerant Control System* (FTCS) merupakan suatu pengembangan teknologi pengendalian untuk kebutuhan keamanan (*safety*) dan perbaikan unjuk kerja. FTCS dapat dibagi menjadi dua tipe, yaitu *Active Fault Tolerant Control System* (AFTCS) dan *Passive Fault Tolerant Control System* (PFTCS) (Zhang & Jiang, 2008). Dalam penelitian sebelumnya, telah dilakukan perancangan sistem AFTC pada pengendalian posisi servo modular MS150 DC menggunakan sistem kendali satu *loop* dengan kesalahan pada sensor dan aktuator (Akbar, 2017).

Mempertimbangkan penelitian-penelitian yang sudah dilakukan sebelumnya, maka pada tugas akhir ini akan dilakukan perancangan sistem *Active Fault Tolerant Control* (AFTC) untuk pengendalian posisi dengan kontrol *cascade* pada sistem servo modular MS150 DC dengan kesalahan pada sensor dan aktuator. Dengan adanya kontrol *cascade*, diharapkan *overshoot* dan osilasi akan berkurang. Selain itu, AFTC diterapkan untuk menolerir kesalahan yang terjadi pada sistem servo modular MS150 DC sehingga sistem dapat bekerja dengan stabil dan performansi tetap terjaga.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, rumusan masalah dari penulisan tugas akhir ini adalah :

1. Bagaimana merancang pengendali PI-P untuk pengendalian posisi MS 150 DC dengan struktur kontrol *cascade* secara *real time*?
2. Bagaimana merancang sistem AFTC untuk struktur kontrol *cascade* pada MS 150 DC yang mengalami kesalahan sensor dan aktuator?

1.3 Tujuan

Tujuan yang ingin dicapai dalam pengerjaan tugas akhir ini adalah membangun sistem pengendalian *cascade* yang dapat

mengakomodasi kesalahan sensor dan aktuator dengan metode AFTC untuk pengendalian posisi MS150 DC

1.4 Batasan Masalah

Berdasarkan identifikasi masalah maka batasan masalah dari tugas akhir ini adalah :

- a. Sistem yang digunakan tanpa penambahan variasi beban (*torque load*).
- b. Variabel yang diukur adalah kecepatan dan posisi motor dalam bentuk tegangan dengan kesalahan sensor hanya terjadi pada pengukuran posisi.
- c. Komponen yang digunakan dalam menjalankan aksi kendali atau aktuator adalah rangkaian *servo amplifier* dengan kesalahan yang terjadi adalah kesalahan bias.
- d. Sistem AFTC yang dirancang diterapkan pada satu kondisi masukan (*setpoint*).

1.5 Sistematika Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

- a. **BAB I PENDAHULUAN**
Pada bab I ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika laporan.
- b. **BAB II TINJAUAN PUSTAKA**
Pada bab II ini dibahas mengenai teori-teori yang berkaitan dengan penelitian yang akan dilakukan, seperti teori identifikasi sistem, pengendalian *cascade*, sistem *Active Fault Tolerant Control*.
- c. **BAB III METODOLOGI PENELITIAN**
Pada bab III ini berisi mengenai rancangan dari penelitian yang dilakukan, metode dan langkah-langkah dalam melakukan penelitian.
- d. **BAB IV ANALISIS DATA DAN PEMBAHASAN**
Pada bab IV ini berisi tentang analisis hasil perancangan sistem *active fault tolerant control* dibandingkan dengan hasil perancangan kontrol konvensional.

e. **BAB V KESIMPULAN DAN SARAN**

Pada bab V ini diberikan kesimpulan tentang tugas akhir yang telah dilakukan berdasarkan analisis data yang diperoleh, serta diberikan saran sebagai penunjang maupun pengembangan tugas akhir selanjutnya.

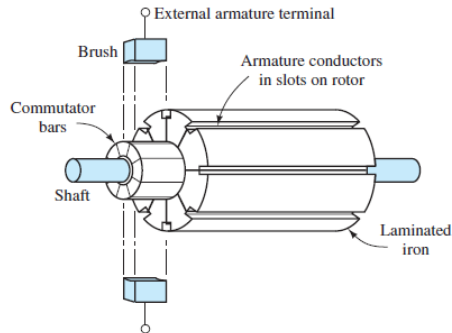
BAB II

TINJAUAN PUSTAKA

2.1 Motor DC

Motor DC merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan jangkar disebut rotor (bagian yang berputar) dan kumparan medan pada motor DC disebut stator (bagian yang tidak berputar). Saat terjadi putaran pada kumparan jangkar, maka akan timbul tegangan yang berubah-ubah arah pada setiap setengah putaran dan menghasilkan tegangan bolak-balik. Arus searah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator sehingga arus yang berbalik arah dengan kumparan jangkar berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen (Sumanto, 1994).

Motor DC magnet permanen tersusun dari beberapa bagian seperti stator, kumparan jangkar atau anker dinamo (*armature*), komutator (*commutator*), dan sikat (*brush*) yang ditunjukkan pada gambar 2.1. Magnet permanen sebagai stator motor tersusun dari dua magnet dengan kutub yang berbeda dan berfungsi menghasilkan fluks yang bernilai konstan dari kutub utara ke kutub selatan. Komutator berfungsi mengumpulkan arus induksi dari jangkar dan mengkonversinya menjadi arus searah. Sikat akan menyalurkan arus listrik dari sumber diluar motor ke dalam kumparan jangkar dan kumparan jangkar (*armature*) sebagai rotor motor digambarkan dalam bentuk sebuah kawat yang memiliki bentuk persegi panjang yang mengubah energi listrik menjadi energi mekanik dalam bentuk gerak putar.



Gambar 2.1. Motor DC (Hambley, 2014)

2.2 Sistem Servo Modular MS150 DC

Sistem servo modular MS150 DC merupakan blok rangkaian elektronik yang digunakan untuk pengendalian kecepatan dan posisi dari motor DC.



Gambar 2.2. Sistem servo modular MS150 DC
(Instruments, 2013)

Sistem servo modular MS150 DC secara umum terdiri dari beberapa bagian yaitu :

a. *Power supply*

Sumber tegangan yang dipakai untuk menyuplai rangkaian pengaturan motor DC adalah modul Feedback tipe PS150E. Alat ini menghasilkan tegangan keluaran sebesar 24V DC

2A ke *servo amplifier*. Pada panel depan terdapat dua set *socket* untuk pasokan tegangan referensi.

b. Motor DC

Motor DC yang digunakan merupakan motor DC tipe DCM150F yang terdiri dari tiga bagian yaitu motor DC magnet permanen dengan poros yang diperpanjang, *tachogenerator* dengan output diatas unit, dan poros berkecepatan rendah.

c. *Servo amplifier*

Modular *feedback servo amplifier* SA150D terdiri dari rangkaian transistor yang dapat menggerakkan motor DC dengan dua arah putaran. Untuk mencegah motor dari *overloading*, terdapat pengukur arus agar arus yang mengalir tidak lebih dari 2A.

d. *Attenuator*

Attenuator AU150B berfungsi sebagai pemberi hambatan yang terdiri dari dua potensio dengan kapasitas masing-masing sebesar 10 K Ω yang dapat digunakan secara bersamaan.

e. *Output Potentiometer*

Potensiometer OP150K berfungsi sebagai sensor posisi. Potensiometer jenis ini merupakan potensiometer jenis rotasi (*rotary*) yang dapat diputar 360 derajat.

f. *Tachogenerator*

Tachogenerator GT150X berfungsi untuk mengkonversi besaran mekanik menjadi besaran listrik sehingga putaran motor dapat diukur.

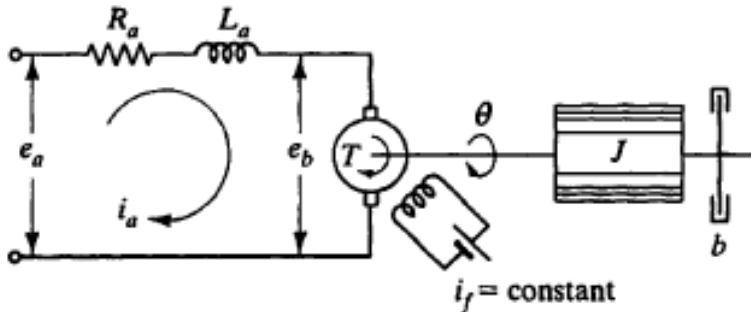
g. *Op Amp Unit*

Operational Amplifier (OA150A) berfungsi sebagai penguat pada pengendalian umpan balik dan dapat menjumlahkan dua atau tiga sinyal.

2.3 Model Matematik Motor DC

Suatu dasar yang digunakan dalam membangun suatu sistem kendali adalah kemampuan untuk memodelkan secara matematika dari sistem yang akan dikendalikan. Model

matematika dari suatu sistem dapat berupa persamaan differensial, fungsi transfer atau ruang keadaan (*state space*). Prinsipnya terdapat dua macam pemodelan matematika yang dapat dipilih, yaitu berdasarkan pada teori pengetahuan dengan menggunakan hukum fisika atau dengan proses eksperimen yaitu melakukan pengukuran. Kebanyakan kasus tidak mungkin untuk membuat model yang sempurna hanya dengan menggunakan pengetahuan fisika saja. Beberapa parameter harus ditentukan dari suatu eksperimen. Model matematik untuk *plant* motor DC dapat dijabarkan pada bagian elektrik dan mekanik seperti yang dijelaskan pada gambar 2.3.



Gambar 2.3. Rangkaian motor DC (Ogata, 2004)

Tegangan untuk sistem motor DC adalah masukan yang dipasang pada armatur (R , L , θ motor). Tegangan tersebut oleh motor diubah menjadi gerak putar yang merupakan keluaran sistem sebagai kecepatan rotasi dari motor. Kecepatan rotasi dinyatakan dengan $d\theta/dt$. Torsi yang dibangkitkan oleh motor DC adalah proporsional terhadap arus motor (armatur) dan kuat medan magnet, dengan asumsi bahwa medan magnet konstan maka torsi motor T proporsional terhadap arus motor i dengan faktor konstanta K_t seperti dituliskan pada persamaan berikut

$$T = K_t \cdot i \quad (2.1)$$

Arus jangkar magnet menghasilkan torsi yang bekerja terhadap inersia dan gesekan sehingga

$$J \frac{d\omega}{dt} + b\omega = T = K.i \quad (2.2)$$

Kecepatan jangkar magnet servomotor DC dikendalikan oleh tegangan kumparan magnet e_a . Persamaan diferensial rangkaian kumparan magnet adalah

$$L \frac{di}{dt} + Ri + e_b = e_a \quad (2.3)$$

$$L \frac{di}{dt} + Ri = V - K\dot{\theta} \quad (2.4)$$

Keterangan:

- L = induktansi kumparan jangkar (H)
- I = arus kumparan jangkar (A)
- R = tahanan kumparan jangkar (Ω)
- V = tegangan masukan servo (V)
- e_a = tegangan yang dikenakan pada jangkar (V)
- e_b = gaya gerak listrik balik (V)

Dengan menggunakan transformasi *Laplace* maka persamaan (2.3) dan (2.4) dapat diubah menjadi persamaan dalam bentuk domain s menjadi

$$s(Js + b)\theta(s) = KI(s) \quad (2.5)$$

$$(Ls + R)I(s) = V - Ks\theta(s) \quad (2.6)$$

Fungsi alih didapatkan dengan mengeliminasi $I(s)$, dimana kecepatan sudut sebagai keluaran dan tegangan sebagai masukan.

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad (2.7)$$

Untuk mendapatkan fungsi alih dimana posisi sebagai keluaran sistem maka dapat dilakukan dengan menginteggralkan $\dot{\theta}$ dengan cara membagi persamaan (2.7) dengan s .

$$\frac{\theta(s)}{V(s)} = \frac{K}{s((Js + b)(Ls + R) + K^2)} \quad (2.8)$$

Keterangan:

- $\theta(s)$ = posisi motor (rad)
- $V(s)$ = tegangan masukan servo (V)
- B = koefisien gesekan viskos ekuivalen dari motor dan beban pada poros motor (N.m/rad.dt)
- K = konstanta torsi motor
- R = tahanan kumparan jangkar (Ω)
- J = momen inersia ekuivalen dari motor (kg.m^2)
- L = induktansi kumparan jangkar (H)

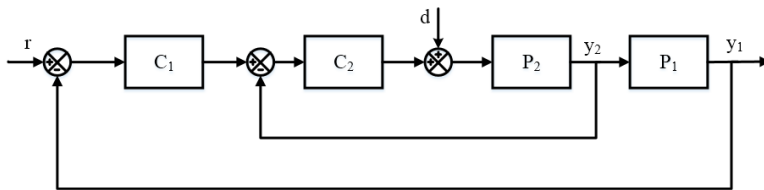
Dalam bentuk persamaan ruang keadaan, persamaan (2.8) dapat dituliskan dengan memilih posisi motor, kecepatan rotasi motor, sebagai variabel *state* dan tegangan sebagai masukan serta posisi motor sebagai keluaran.

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-b}{J} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} V \quad (2.9)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad (2.10)$$

2.4 Sistem Pengendalian *Cascade*

Sistem pengendalian adalah gabungan dari kerja suatu komponen yang digunakan untuk mempertahankan variabel yang dikendalikan (*process variable*) pada suatu nilai tertentu (*set point*) sehingga sistem dapat dikatakan stabil. Suatu sistem linear (LTI/Linear Time Invariant) dikatakan stabil jika respon natural sistem tersebut mendekati nol pada waktu mendekati tak terhingga. Sistem LTI dikatakan tidak stabil jika respon natural meningkat tanpa batas jika waktu mendekati tak terhingga. Salah satu jenis sistem pengendalian yang biasa digunakan adalah sistem pengendalian *cascade*.



Gambar 2.4. Skema pengendalian cascade (Visioli, 2006)

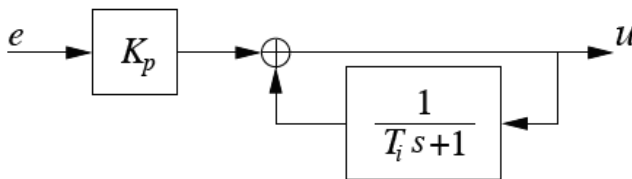
Sistem kontrol *cascade* adalah salah satu jenis strategi kontrol yang digunakan untuk mengoptimasi suatu sistem kontrol. Strategi ini digunakan untuk mengurangi *deviasi* dan *error* yang disebabkan oleh *disturbance* dari system. Ciri khas sistem pengendalian *cascade* (bertingkat) adalah adanya *manipulated variable* (variabel yang dimanipulasi) sebuah pengendali yang menjadi *set point* dari pengendali lain. ada dua jalur umpan balik pada sistem pengendalian bertingkat (*cascade control*), sehingga terbentuk dua mata rantai pengendalian (kalang). Mata rantai atau kalang bagian luar (*outer loop*) disebut *primary loop* atau *master*, dan mata rantai atau kalang bagian dalam (*inner loop*) disebut *secondary loop* atau *slave*. *Master* atau *primary loop* mengendalikan proses variabel primer sedangkan *slave* atau *secondary loop* mengendalikan proses variabel sekunder. Jika $P1(s)$ mewakili dinamika proses yang lambat dan $P2(s)$ mewakili dinamika proses yang cepat, keefektifan sistem kontrol cascade ini karena fakta bahwa gangguan yang mempengaruhi loop

sekunder secara efektif dikompensasi sebelum mempengaruhi output pada proses utama (Visioli, 2006).

2.5 Sistem Kendali PI-P

Algoritma pengendali *Proportional-Integral* saat ini banyak diaplikasikan dalam pengontrolan beragam variabel proses industri. Karakteristik pengendali PI- P sangat dipengaruhi oleh kontribusi besaran dari nilai parameter P dan I. Penyetelan konstanta K_p dan T_i akan mengakibatkan menonjolan sifat dari masing – masing elemen.

Pengendali *proportional* memiliki keluaran yang sebanding atau proporsional dengan besar sinyal *error*. Jadi, keluaran pengendali *proportional* merupakan perkalian antara konstanta/gain *proportional* dengan masukannya yang berupa sinyal *error*. Pengendali PI merupakan penggabungan antara dua macam pengendali, yaitu P (*Propotional*) dan I (*Integral*). Pengendali PI merupakan jenis pengendali untuk menentukan suatu kestabilan atau kepresisian sistem instrumentasi dengan adanya *feedback* atau umpan balik pada sistem tersebut. Diagram blok sistem pengendali PI adalah sebagai berikut :



Gambar 2.5. Diagram blok pengendalian PI (Visioli, 2006)

Pengendali P berfungsi untuk mempercepat *rise time* agar respon sistem lebih cepat untuk mencapai *setpoint*, akan tetapi pengendali ini mempunyai kekurangan yaitu meningkatkan *offset*. Persamaan pengendali P dalam bentuk diskrit adalah sebagai berikut:

$$u(z) = K_p \cdot e(z) \quad (2.11)$$

dengan

$u(z)$: sinyal kontrol dalam bentuk diskrit

$e(z)$: sinyal *error* dalam bentuk diskrit

K_p : *gain* proportional

Kelemahan dari pengendali P dapat dihilangkan dengan pengendali I sehingga dapat menghilangkan *offset* dan mengurangi *maksimum overshoot* yang terlalu besar.

Persamaan pengendali I dalam diskrit adalah sebagai berikut :

$$u(z) = \frac{K_i T_s}{\tau_i (z-1)} e(z) \quad (2.12)$$

dimana

$u(z)$: sinyal kontrol dalam bentuk diskrit

$e(z)$: sinyal *error* yang merupakan selisih antara *setpoint* dan keluaran proses dalam bentuk transformasi z

K_i : *gain* integral

T_s : waktu cuplik

Pengendalian PI menghasilkan respon yang lebih cepat dari pengendali I tapi mampu menghilangkan *offset* yang ditinggalkan pengendali P.

Persamaan pengendali PI adalah sebagai berikut :

$$u(z) = K_p e(z) + \frac{K_i T_s}{\tau_i (z-1)} e(z) \quad (2.13)$$

Jika pemodelan matematis sistem susah untuk dilakukan. maka perancangan pengendali PI-P secara analitis tidak mungkin dilakukan, sehingga perancangan pengendali PI-P harus dilakukan secara eksperimental.

Beberapa parameter dalam algoritma PI-P yang harus diketahui dalam suatu sistem adalah:

- a. *Maximum (percent) overshoot* adalah nilai maksimum respon sistem pengukuran terhadap keadaan yang diinginkan. Jika nilai *steady state* dari respon berbeda dengan nilai yang diinginkan, umumnya menggunakan *maximum percent overshoot* dengan persamaan :

$$\text{Maximum percent overshoot} = \frac{y_p - y_{ss}}{y_{ss}} \times 100\% \quad (2.14)$$

dengan y_p adalah nilai maksimum respon system dan y_{ss} adalah nilai *steady state* sistem.

- b. *Settling time* adalah waktu yang dibutuhkan respon sistem untuk mencapai dan berada direntang $\pm 5\%$ atau $\pm 2\%$ dari nilai akhir.
- c. *Error steady state* adalah nilai *error* yang dihasilkan dari keadaan *steady* ketika belum mencapai *set point* yang diinginkan.

2.6 Sistem Active Fault Tolerant Control (AFTC)

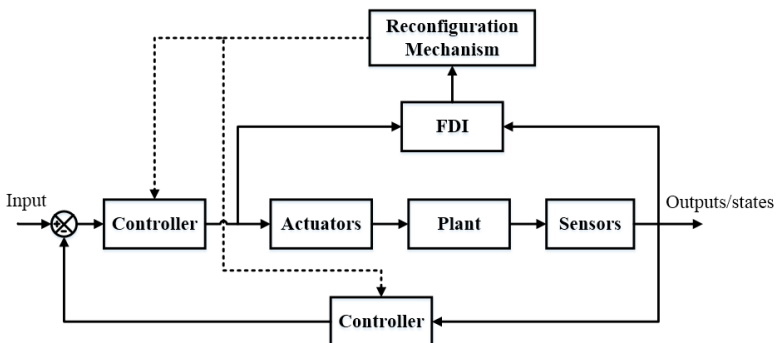
Peningkatan performansi dan *safety* merupakan kunci modern sistem teknologi, yang dipenuhi oleh skema kontrol yang canggih. Skema kontrol yang dirancang secara konvensional dalam sistem dapat menghasilkan kinerja yang tidak memuaskan atau bahkan ketidakstabilan yang terjadi di dalam komponen sistem seperti sensor dan aktuator. Untuk meningkatkan kehandalan sistem, dilakukan desain pengendalian yang mampu menoleransi kerusakan yang terjadi dengan tetap menjaga stabilitas sistem yang diinginkan. Sistem *Fault Tolerant Control* (FTC) adalah sebuah pengendali yang mampu mentoleransi kesalahan dalam suatu sistem untuk meningkatkan kinerja yang diinginkan (Tushar Jain, 2017)

FTC dapat diklasifikasikan pada 2 tipe yaitu sistem *Passive Fault Tolerance Control* (PFTC) dan sistem *Active Fault Tolerance Control* (AFTC). Sistem PFTC didesain untuk menjadi sistem pengendali yang *robust* dari kesalahan suatu komponen.

Sedangkan sistem AFTC bereaksi terhadap kesalahan yang terjadi pada suatu komponen dengan merekonfigurasi aksi pengendali sehingga kestabilan dan kinerja pada sistem dapat dijaga. Dengan demikian, tujuan utama dari sistem FTC adalah merancang pengendali dengan struktur yang cocok untuk mencapai kestabilan dan kinerja yang diinginkan. Tidak hanya ketika komponen pengendali berfungsi secara normal tetapi ketika terjadi kesalahan pada suatu komponen tersebut.

Active Fault Tolerance Control (AFTC) secara umum terdiri atas:

- a. *Fault Detection Diagnosis (FDD)* atau *Fault Detection and Identification (FDI)*
- b. *Reconfigurable Control (RC)*



Gambar 2.6. Struktur umum sistem AFTC (Mahmoud, 2003)

Fault Detection and Diagnosis (FDD) merupakan suatu metode untuk mendeteksi *fault* atau kesalahan yang terjadi pada aktuator, sensor, maupun komponen penyusun lain dalam suatu sistem. Tujuan dari pendeteksian kesalahan ini agar sinyal pengendali yang baru dapat menyampaikan informasi pada pengendali untuk menoleransi kesalahan yang terjadi. Dalam FDD terdapat tiga hal yang dilakukan yaitu mendeteksi kesalahan pada sistem serta waktu selang selama terjadinya kesalahan, mengisolasi lokasi kesalahan dan tipe kesalahan yang terjadi dan mengidentifikasi besarnya kesalahan yang terjadi. *Observer* merupakan salah satu jenis FDD yang berfungsi untuk

mengestimasi kesalahan berdasarkan model matematik (Indriawati, dkk, 2013)

Reconfigurable Control berfungsi untuk memulihkan sistem yang berada di bawah kesalahan agar dapat mencapai tujuan pengendalian yang diinginkan. *Reconfigurable control* harus dapat bekerja secara otomatis untuk menjaga stabilitas dan performansi yang diinginkan. Untuk merancang sistem kendali yang toleran terhadap kesalahan aktuator dan sensor, maka dilakukan konfigurasi ulang sinyal pengendali atau *reconfigurable control* secara otomatis berdasarkan jenis kesalahan yang terjadi. Konfigurasi ulang sinyal kendali untuk mengatasi kesalahan pada aktuator dan sensor memerlukan proses diagnosis kesalahan (*fault diagnosis*) untuk mendeteksi dan mengestimasi kesalahan yang terjadi.

2.7 Observer

Observer merupakan suatu metode yang digunakan untuk mengestimasi keadaan sistem, *noise* serta kesalahan pada sensor dan aktuator. Pemodelan suatu sistem dapat diubah dalam bentuk *state space* dalam diskrit seperti dibawah ini:

$$x(k+1) = Ax(k) + Bu(k) \quad (2.15)$$

$$y(k) = Cx(k) + Du(k) \quad (2.16)$$

dengan:

A = matriks keadaan (*state*)

B = matriks kendali

C = matriks keluaran

D = matriks gangguan

$x(k)$ = vektor keadaan (*state*)

$x(k+1)$ = turunan vektor keadaan

$u(k)$ = masukan (*input*) kendali

$y(k)$ = vektor keluaran

Model matematis dari *observer* sama seperti model matematis pada sistem dengan ditambahkan K_e (*observer gain matrix*) menjadi persamaan sebagai berikut

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K_e(y(k) - \hat{y}(k)) \quad (2.17)$$

$$\hat{y}(k) = C\hat{x}(k) \quad (2.18)$$

dengan :

$\hat{x}(k+1)$ = Estimasi state $x(k+1)$

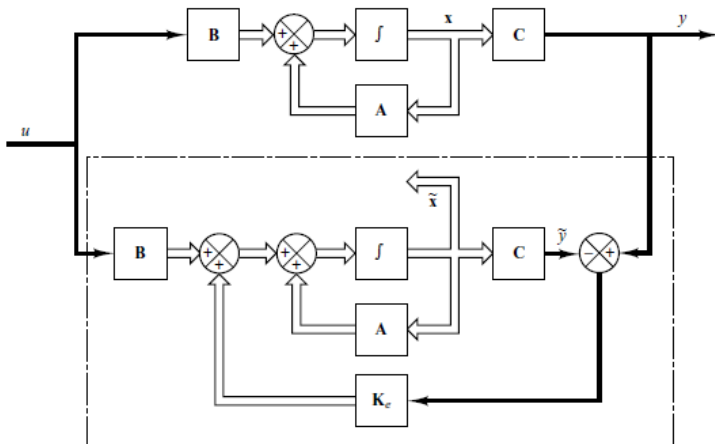
K_e = Gain Observer

$\hat{x}(k)$ = Estimasi state $x(k)$

$\hat{y}(k)$ = Estimasi $y(k)$

$y(k)$ pada persamaan (2.16) dan $\hat{y}(k)$ pada (2.18) disubstitusikan ke dalam persamaan (2.17), maka diperoleh

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K_e(Cx(k) - C\hat{x}(k)) \quad (2.19)$$



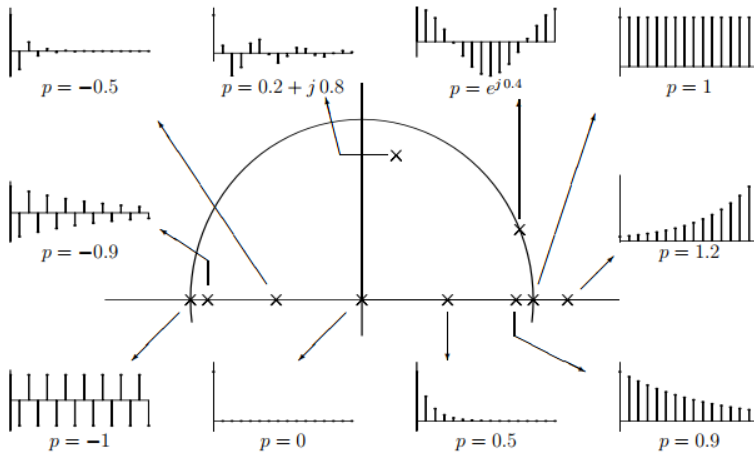
Gambar 2.7. *Observer* (Ogata, 2010)

Berdasarkan gambar 2.7, *observer* mengandung nilai *gain observer* atau K_e . *Gain observer* merupakan matriks untuk mengoreksi perbedaan antara keluaran y aktual dan y estimasi sehingga dapat digunakan untuk mengestimasi *state* yang tidak terukur. Untuk menentukan nilai K_e sangat dipengaruhi oleh nilai letak *pole*, seperti pada gambar 2.8 untuk sistem dengan domain diskrit, posisi nilai *pole* untuk menyatakan sistem stabil berada tidak lebih dari 1 atau -1 pada sumbu imajiner maupun sumbu riil. Kemudian persamaan karakteristik observer menjadi

$$|zI - A + K_e C| = 0 \quad (2.20)$$

Dengan menentukan nilai p (posisi *pole*), diperoleh persamaan karakteristik yang diinginkan, maka nilai K_e diperoleh dari menyelaraskan persamaan (2.20)

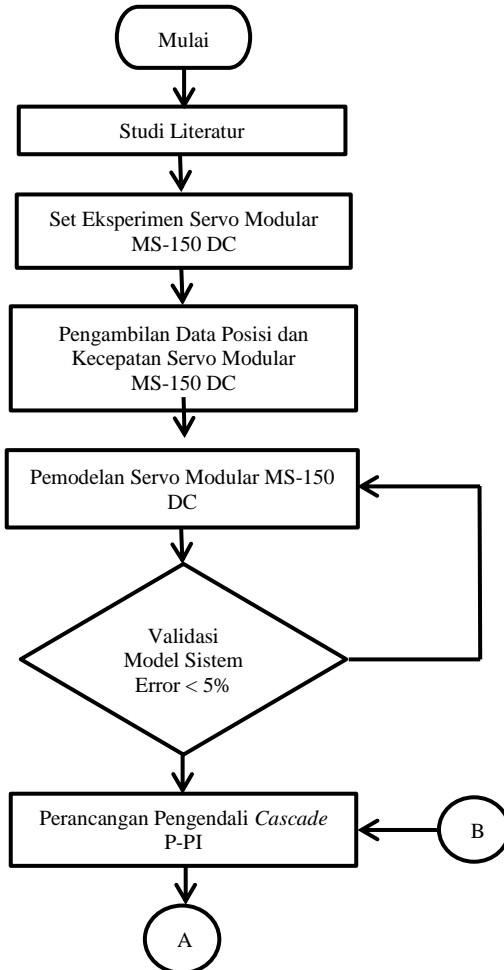
$$|zI - A + K_e C| = (z - p)(z - p) \dots (z - p_n) \quad (2.21)$$

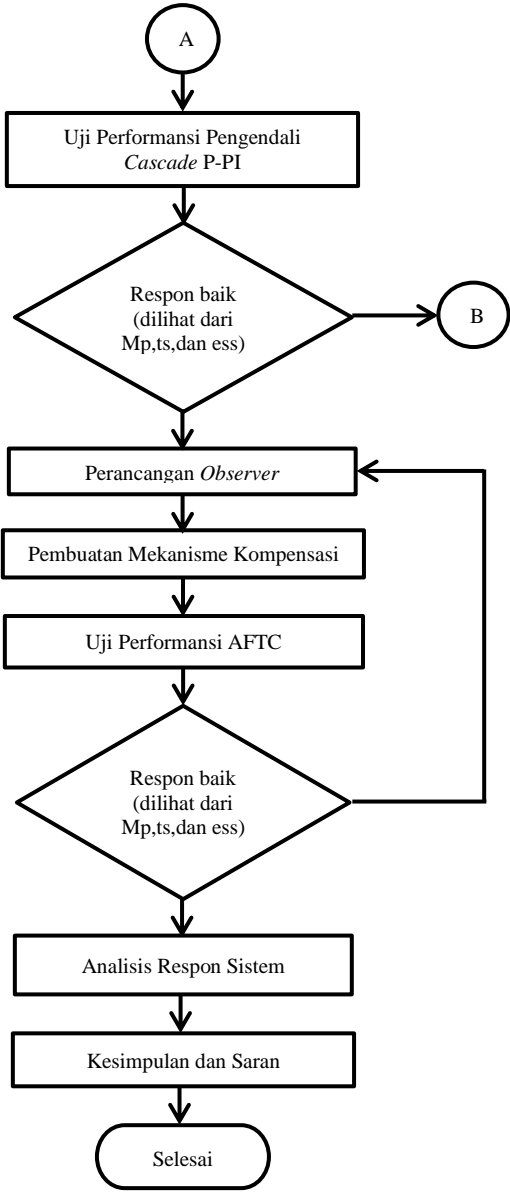


Gambar 2.8. Kestabilan *pole* sistem diskrit (Boom, 2006)

BAB III METODOLOGI PENELITIAN

Tahapan yang dilakukan dalam Tugas Akhir ini ditampilkan dengan sebuah diagram alir (*flowchart*) pada gambar 3.1





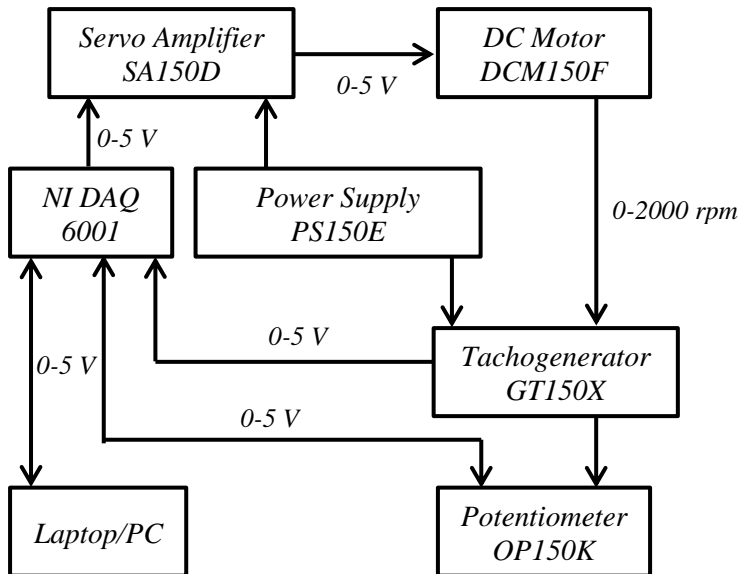
Gambar 3.1. Diagram alir penelitian tugas akhir

3.1 Studi Literatur

Studi literatur dilakukan untuk pemahaman awal secara teoritis terhadap materi pendukung pada penelitian tugas akhir ini antara lain adalah pemahaman sistem servo modular MS150 DC, identifikasi sistem, sistem pengendalian *cascade* dan sistem kendali PID, dan sistem *Active Fault Tolerant Control (AFTC)*.

3.2 Set Eksperimen Sistem Servo Modular MS150 DC

Pengendalian posisi dapat dilakukan dengan mengatur sinyal masukan tegangan pada motor DC. Pada motor DC, maka perlu dirancang seperti gambar berikut ini



Gambar 3.2. Diagram set eksperimen sistem MS150 DC

Sinyal masukan yang diberikan ke motor DC berupa tegangan dengan rentang 0-5 V. Kemudian perputaran motor 0-2000 rpm dideteksi dengan *tachometer* yang kemudian dikonversi menjadi tegangan 0-5 V dan digunakan untuk sinyal masukan DAQ NI-6001 sebagai variable proses. Perintah dari PC diberikan

kepada DAQ sebagai sinyal kendali untuk servo dimana tegangan keluaran DAQ diatur dengan rentang 0-5 volt. Tegangan terminal yang berupa sinyal kendali merupakan masukan bagi *servoamplifier*. Masukan tersebut akan menggerakkan motor DC, dimana tegangan terminal akan proporsional dengan kecepatan.



Gambar 3.3. Set eksperimen *real plant* sistem MS150 DC

Hasil pergerakan dari motor DC akan dikopel dengan *tachogenerator*. Pergerakan pada bagian kecepatan rendah akan dibaca oleh potensiometer yang kemudian dikonversi ke besaran tegangan dan masuk dalam PC/laptop.

3.3 Pemodelan Servo Modular MS 150 DC

Pemodelan dilakukan dengan pemilihan model berupa persamaan fungsi transfer. Pemilihan fungsi transfer model harus mendekati grafik pengukuran dari *real plant*. Studi literatur mengenai model untuk kecepatan pada motor DC digunakan sebagai dasar memilih orde denominator dan numerator pada program *labVIEW* 2013.

Pemodelan kecepatan dengan menggunakan perangkat lunak *labVIEW* 2013 menghasilkan nilai fungsi transfer dalam domain

z. Sistem yang dibuat merupakan sistem dengan domain diskrit berupa persamaan dalam bentuk fungsi transfer orde 1.

Persamaan fungsi transfer orde 1 untuk kecepatan motor DC memiliki nilai sebagai berikut :

$$\text{Kecepatan} = \omega(z) = \frac{0.91}{z - 0.5948} u(z) \quad (3.1)$$

Persamaan (3.1) kemudian dirubah kedalam persamaan dengan domain k yang selanjutnya dirubah ke dalam bentuk persamaan ruang keadaan (*state space*), yang dituliskan :

$$x(k+1) = Ax(k) + Bu(k) \quad (3.2)$$

$$y(k) = Cx(k) + Du(k) \quad (3.3)$$

a. Kecepatan

$$\omega(z) = \frac{0.91}{z - 0.5948} u(z) \quad (3.4)$$

$$\omega(z)(z - 0.5948) = 0.91u(z) \quad (3.5)$$

$$z\omega(z) - 0.5948\omega(z) = 0.91u(z) \quad (3.6)$$

$$z\omega(z) = 0.5948\omega(z) + 0.91u(z) \quad (3.7)$$

persamaan (3.7) di transformasi balik, menjadi persamaan

$$\omega(k+1) = 0.5948\omega(k) + 0.91u(k) \quad (3.8)$$

b. Posisi

$$\frac{\theta(z)}{\omega(z)} = \frac{K_x T_s}{z - 1} \quad (3.9)$$

$$\theta(z)(z - 1) = K_x T_s \omega(z) \quad (3.10)$$

Kemudian persamaan (3.10) diinvers Z menjadi persamaan

$$\theta(k+1) - \theta(k) = K_x T_s \omega(k) \quad (3.11)$$

$$\theta(k+1) = \theta(k) + K_x T_s \omega(k) \quad (3.12)$$

Diketahui,

$$K_x = 0.6$$

$$T_s = 0.1 \text{ detik}$$

Maka didapatkan,

$$\theta(k+1) = \theta(k) + K_x T_s \omega(k) \quad (3.13)$$

$$\theta(k+1) = \theta(k) + 0,06\omega(k) \quad (3.14)$$

Persamaan (3.8) dan (3.14) kemudian dibentuk menjadi persamaan ruang keadaan untuk sistem servo modular MS150 DC dalam domain k sebagai:

$$\begin{bmatrix} \theta(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0,06 \\ 0 & 0,5948 \end{bmatrix} \begin{bmatrix} \theta(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0,91 \end{bmatrix} u(k) \quad (3.15)$$

$$y(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta(k) \\ \omega(k) \end{bmatrix} \quad (3.16)$$

3.4 Validasi Pemodelan

Validasi pemodelan dari motor DC, *tachogenerator* dan *servoamplifier* dilakukan untuk memastikan model yang diperoleh sudah merepresentasikan sistem dalam kondisi riil.

Validasi dilakukan dengan cara membandingkan respon antara sistem riil dengan model fungsi transfer dari sistem. Dengan memasukkan nilai masukan/*input* 0.2 volt, diperoleh respon *real* dan model sistem. Untuk mengetahui nilai *error* antara respon sistem *real* dan model, digunakan persamaan sebagai berikut :

$$MAPE = \frac{\sum_{t=1}^n |e_t| / F_t}{n} \times 100 \quad (3.17)$$

dengan

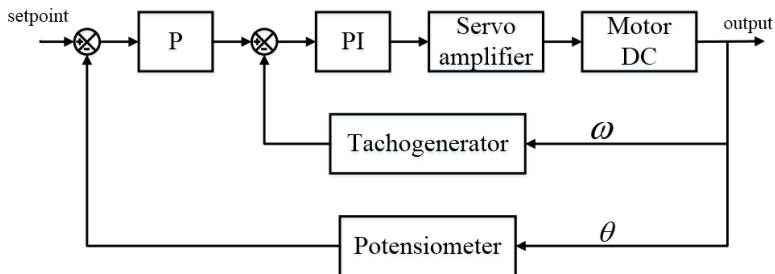
e_t = error model (nilai aktual - nilai pemodelan) pada

periode t
 F_t = nilai pemodelan pada periode t
 n = jumlah data

MAPE merupakan pengukuran kesalahan yang menghitung ukuran presentase penyimpangan antara data aktual dengan data pemodelan. Batas *error* validasi maksimal yang digunakan sebesar 5%. Semakin kecil nilai *error* yang diperoleh maka semakin baik hasil pemodelan dan hasil model yang diperoleh sudah cukup mereprsentasikan sistem dalam kondisi riil.

3.5 Perancangan Sistem Kendali *Cascade* PI-P

Diagram blok sistem pengendalian posisi sistem servo modular MS150 DC ditunjukkan oleh gambar berikut



Gambar 3.4. Diagram pengendalian *cascade* P-PI

Karakteristik pengendali P-PI sangat dipengaruhi oleh kontribusi besaran dari parameter P dan I. Nilai K_p dan K_i didapatkan dengan metode *trial and error* untuk perancangan secara simulasi dengan *simulink* pada perangkat lunak *matlab* R2103a, kemudian setelah sistem kendali dapat bekerja dengan baik, dilakukan perancangan sistem kendali *cascade* P-PI untuk diterapkan secara *real time* pada sistem servo modular MS150 DC dengan menggunakan perangkat lunak *labVIEW* 2013. Nilai *gain* K_p dan K_i untuk simulasi terkadang tidak tepat untuk digunakan secara *real time* jika perangkat lunak yang digunakan

berbeda, sehingga diperlukan perubahan nilai *gain* yang sesuai untuk diterapkan secara *real time*.

Perancangan sistem kendali P-PI bertujuan untuk mengetahui respon dari sistem jika ditambahkan dengan pengendali. Sistem yang dirancang telah bekerja dengan baik jika respon sistem dapat mencapai nilai *setpoint* yang telah diberikan.

3.6 Perancangan Sistem *Active Fault Tolerant Control (AFTC)* Secara Simulasi

3.6.1 Perancangan *Observer*

Perancangan *observer* dilakukan untuk mengestimasi kesalahan yang ada, dengan cara mengembangkan persamaan ruang keadaan sistem. Persamaan untuk perancangan *observer* diperoleh dari penelitian yang telah dilakukan tetapi perubahan dilakukan dari bentuk domain kontinu ke dalam bentuk domain diskrit. Nilai matriks ruang keadaan untuk perancangan *observer* diperoleh dari pemodelan sistem servo modular MS150 DC pada persamaan (3.13) dan (3.14) dituliskan kembali sebagai berikut :

$$\begin{bmatrix} \theta(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0,06 \\ 0 & 0,5948 \end{bmatrix} \begin{bmatrix} \theta(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0,91 \end{bmatrix} u(k) \quad (3.18)$$

$$y(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta(k) \\ \omega(k) \end{bmatrix} \quad (3.19)$$

$$A = \begin{bmatrix} 1 & 0,06 \\ 0 & 0,5948 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0,91 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$F_a = \begin{bmatrix} 0 \\ 0,91 \end{bmatrix}$$

Persamaan ruang keadaan yang mengandung kesalahan aktuator dan sensor dituliskan sebagai berikut :

$$x(k+1) = Ax(k) + Bu(k) + F_a f_a(k) \quad (3.20)$$

$$y(k) = Cx(k) \quad (3.21)$$

dengan, $x(k) \in R^n$, $u(k) \in R^m$, $y(k) \in R^p$, $f_a(k) \in R^r$, berturut-turut adalah keadaan (*state*), masukan(output), keluaran(output), kesalahan aktuator dan kesalahan sensor. Sedangkan A, B, C, D dan F_a adalah matriks keadaan, masukan, keluaran, dan kesalahan aktuator.

Setelah nilai-nilai matriks didapatkan, persamaan keadaan dan persamaan keluaran yang mengandung kesalahan aktuator dan digunakan untuk merancang persamaan *observer* yang dituliskan:

$$\hat{x}(k+1) = A\hat{x} + B.u(k) + F_a.f(k) + K_e(y(k) - \hat{y}(k)) \quad (3.22)$$

$$\hat{y}(k) = C\hat{x}(k) \quad (3.23)$$

dengan K_e adalah gain observer. Kemudian ditambahkan didefinisikan variable baru \hat{f}_a untuk mengestimasi kesalahan aktuator dengan menggunakan persamaan berikut:

$$\hat{f}_a(k+1) = T_s L_e(k) + \hat{f}_a(k) \quad (3.24)$$

dengan L_e adalah gain observer untuk kesalahan aktuator sehingga dari persamaan (3.22) dan (3.24) digabungkan kedalam bentuk yang lebih ringkas, menjadi

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{f}_a(k+1) \end{bmatrix} = \begin{bmatrix} A & F_a \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{f}_a(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} K_e \\ L_e T_s \end{bmatrix} e(k)$$

$$+ \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} K_e \\ L_e T_s \end{bmatrix} e(k) \quad (3.25)$$

$$\hat{y}(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{f}_a(k) \end{bmatrix} \quad (3.26)$$

dengan $e(k) = y(k) - \hat{y}(k)$. Persamaan (3.30) dan (3.31) dapat dirubah menjadi persamaan keadaan dan persamaan keluaran *observer*, sebagai berikut :

$$\hat{\tilde{x}}(k+1) = \tilde{A}.\hat{\tilde{x}}(k) + \tilde{B}.u(k) + \tilde{K}_e(\tilde{y}(k) - \hat{\tilde{y}}(k)) \quad (3.27)$$

$$\hat{\tilde{y}}(k) = \tilde{C}.\hat{\tilde{x}}(k) \quad (3.28)$$

dengan:

$$\hat{\tilde{x}}(k+1) = \begin{bmatrix} \hat{x}(k+1) \\ \hat{f}_a(k+1) \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} A & F_a \\ 0 & I \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

$$\tilde{K}_e = \begin{bmatrix} K_e \\ L_e T_s \end{bmatrix}$$

selanjutnya yaitu substitusi nilai matriks A, B, C, F, T_s dan I sehingga diperoleh nilai matriks sebagai berikut :

$$\tilde{A} = \begin{bmatrix} 1 & 0,06 & 0 \\ 0 & 0,5948 & 0,91 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} 0 \\ 0,91 \\ 0 \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Untuk memperoleh nilai gain observer (\tilde{K}_e) digunakan metode *pole-placement*. Nilai pole yang diinginkan ditentukan dengan metode *trial-and-error*, dengan berdasarkan ilmu kestabilan berdasarkan letak *pole*. Sebuah sistem diskrit dianggap stabil jika seluruh akar-akar karakteristik bagian riil atau imajiner bernilai tidak lebih dari 1 atau -1 sehingga dengan metode *trial-and-error* diperoleh tiga nilai pole yaitu 0,41; 0,41; 0,42.

Nilai matriks gain observer didapatkan dengan menggunakan program perangkat lunak *matlab* R2013a sehingga diperoleh matriks \tilde{K}_e sebagai berikut :

$$\tilde{K}_e = \begin{bmatrix} 0,59 & 0 & 0 \\ 0,06 & 0,7648 & 0,376 \end{bmatrix}$$

Selanjutnya, estimasi kesalahan sensor diperoleh dengan persamaan berikut:

$$\hat{f}_s = \hat{\theta} - \theta \quad (3.29)$$

dengan

$\hat{\theta}$ = posisi estimasi setelah diberi kesalahan

θ = posisi awal

Setelah dilakukan perancangan *observer* secara simulasi menggunakan perangkat lunak *matlab* R2013a dan *observer* dapat mengestimasi kesalahan dengan benar, maka selanjutnya

dilakukan penerapan *observer* secara *real time* dengan menggunakan perangkat lunak *labVIEW* 2013.

3.6.2 Pembuatan Mekanisme Kompensasi

Reconfigurasi Control atau pengaturan ulang dari sistem kendali dengan metode kompensasi digunakan untuk menoleransi kesalahan pada sensor atau aktuator sehingga sistem tetap dapat beroperasi sesuai *setpoint*. Persamaan kompensasi pada kesalahan aktuator adalah

$$u_m(k) = u(k) - \hat{f}_a(k) \quad (3.30)$$

dengan $u_m(k)$ adalah nilai sinyal kendali yang mengandung kesalahan, $u(k)$ adalah nilai sinyal kendali sebenarnya, $\hat{f}_a(k)$ adalah estimasi kesalahan aktuator. Persamaan kompensasi pada kesalahan sensor adalah

$$y(k) = y_m(k) - \hat{f}_s(k) \quad (3.31)$$

dengan $y(k)$ adalah nilai pengukuran sebenarnya, $y_m(k)$ adalah nilai pengukuran yang mengandung kesalahan, $\hat{f}_s(k)$ adalah estimasi kesalahan sensor. Jika mekanisme kompensasi yang dirancang secara simulasi sudah dapat memperbaiki kesalahan aktuator dan sensor yang terjadi, kemudian diterapkan secara *real time* pada sistem servo modular MS150 DC dengan menggunakan perangkat lunak *labVIEW* 2013 yang dapat dilihat pada lampiran.

3.7 Perancangan Sistem *Active Fault Tolerant Control* (AFTC) Secara *Real Time*

Perancangan AFTC untuk kesalahan pada aktuator dan sensor secara *real time* diberikan melalui perangkat lunak *labVIEW* 2013 dan komponen sistem servo modular MS150 DC secara langsung. Untuk kesalahan pada aktuator diberikan pada *servo amplifier* dan kesalahan sensor diberikan pada

potensiometer di servo modular MS150 DC dengan menggunakan bantuan *attenuator*. Prinsip *attenuator* adalah sebagai pembagi tegangan sebelum masuk ke *servo amplifier* dan setelah keluar dari *potensiometer*.

“Halaman ini memang dikosongkan”

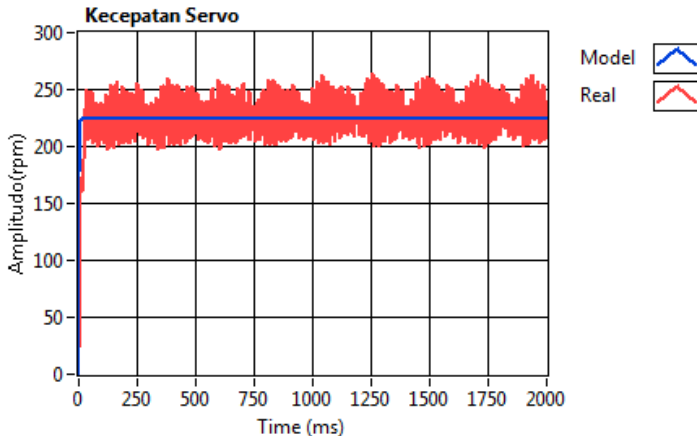
BAB IV

ANALISIS HASIL DAN PEMBAHASAN

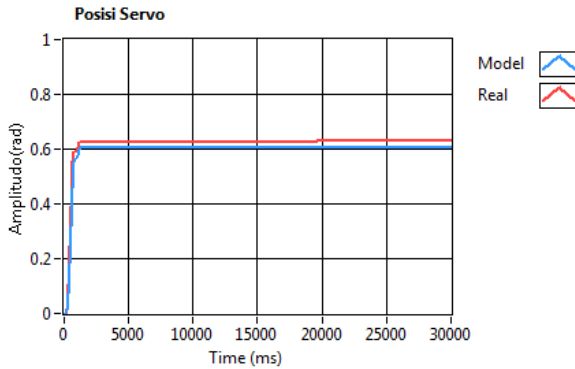
4.1 Validasi Data

Pemodelan sistem menghasilkan persamaan fungsi alih dari kecepatan sistem servo modular MS150 DC. Fungsi alih tersebut selanjutnya dilakukan validasi untuk mengetahui apakah model yang diperoleh telah sesuai dengan kondisi *real* dari sistem servo modular MS150 DC. Validasi dilakukan dengan membandingkan grafik model dan *real* dari kecepatan dan posisi sistem servo modular MS150 DC.

Gambar 4.1 menunjukkan nilai *error* validasi kecepatan sebesar 2,4%, sedangkan gambar 4.2 menunjukkan nilai *error* validasi posisi sebesar 1,7%. Grafik respon kecepatan dan posisi antara model dan sistem secara *real time* sudah saling mendekati sehingga model yang diperoleh sudah dapat merepresentasikan sistem secara *real time*.



Gambar 4.1. Respon sistem validasi kecepatan motor DC



Gambar 4.2. Respon sistem validasi posisi motor DC

4.2 Sistem Kendali *Cascade* PI-P

Perancangan sistem kendali *Cascade* PI-P dilakukan untuk mendapatkan nilai *gain* pengendali yang selanjutnya hasil respon dapat diperoleh sehingga dapat diketahui kinerja dari sistem kendali yang dirancang.

4.2.1 Pengendalian *Cascade* PI-P Secara Simulasi

Dengan menggunakan *simulink* pada perangkat lunak *matlab* R2013a, diperoleh nilai pengendalian sebagai berikut :

-*Inner loop*

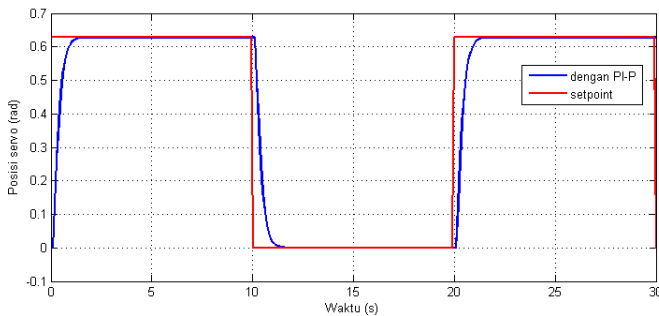
$$K_p = 1$$

$$K_i = 4$$

-*Outer loop*

$$K_p = 3,7$$

Grafik respon sistem kendali PI-P pada motor DC secara simulasi ditunjukkan pada gambar 4.3



Gambar 4.3. Respon sistem perancangan sistem kendali PI-P secara simulasi

Gambar 4.3 menunjukkan 2 grafik, yaitu grafik berwarna merah yang menunjukkan nilai *set point* dan grafik berwarna biru yang menunjukkan hasil pengendalian *cascade*. Nilai *setpoint* sebesar 0,628 radian diberikan sehingga dari kedua grafik tersebut dapat diketahui bahwa sistem dengan PI-P yang dirancang dapat bekerja dengan baik karena mampu mencapai nilai *setpoint*, dengan karakteristik respon nilai *settling time* sebesar 2,2 detik dan tidak ada *overshoot* serta *error steady state*.

4.2.2 Pengendalian *Cascade* PI-P Secara Real

Untuk melihat respon sistem pengendalian posisi motor DC secara langsung (*real time*) maka dilakukan pengendalian dalam perangkat lunak *labVIEW* 2013. Nilai pengendali yang didapat secara eksperimental adalah

-*Inner loop*

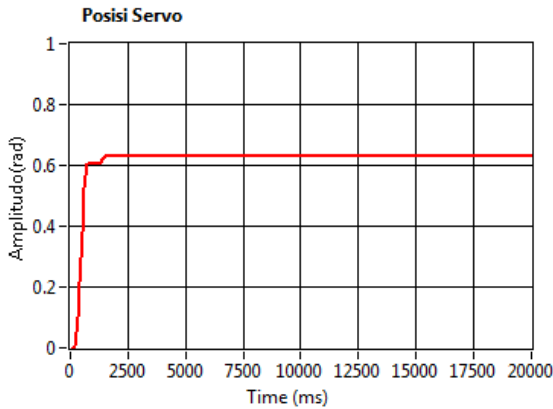
$K_p = 0,7$

$T_i = 0,001$

-*Outer loop*

$K_p = 0,34$

Grafik respon sistem posisi motor DC dengan menggunakan perangkat lunak *labVIEW* 2013 setelah diberi pengendalian dan nilai *setpoint* 0,628 radian ditunjukkan pada gambar 4.4 berikut ini



Gambar 4.4. Respon sistem penerapan sistem kendali PI-P secara *real time*

Gambar 4.4 menunjukkan grafik respon sistem secara *real* untuk kendali PI-P dengan *set point* yang diberikan sebesar 0,628 radian. Dari grafik tersebut dapat diketahui bahwa sistem kendali PI-P dapat berkerja dengan baik karena mampu mencapai nilai *set point* dengan *settling time* sebesar 2,2 detik serta tidak ada *overshoot* dan *error steady-state*.

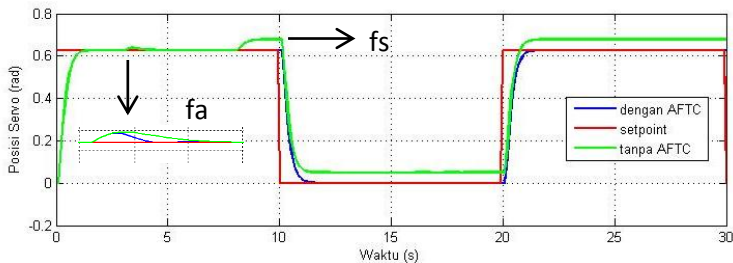
4.3 Uji Performansi

Uji performansi dilakukan untuk menguji algortima *active fault tolerant control* yang telah dibuat dapat memperbaiki kesalahan yang terjadi pada aktuator dan sensor. Uji secara simulasi dilakukan dengan memberikan kesalahan pada aktuator dan sensor berupa yaitu kesalahan bias. Kesalahan bias secara simulasi pada aktuator diberikan sebesar 16% dan 63,7% dari sinyal kendali, sedangkan kesalahan bias pada sensor secara simulasi diberikan sebesar 8% dan 63,7% dari sinyal pengukuran. Kesalahan bias merupakan penyimpangan nilai dari nilai tetapan yang telah ditentukan. Kesalahan bias pada sensor posisi dapat berupa adanya zero bias, yaitu perubahan nilai zero posisi pada sensor *potensiometer*, sedangkan kesalahan bias pada aktuator dapat berupa gangguan yang menghambat tegangan dari *servo*

amplifier untuk masuk ke motor DC. Uji performansi dilakukan secara simulasi dengan menggunakan perangkat lunak *matlab* R2013a untuk mengetahui apakah algoritma AFTC yang dirancang sudah mampu bekerja dengan baik. Selanjutnya uji performansi sistem AFTC dilakukan secara *real time* pada sistem servo modular MS150 DC dengan kesalahan yang terjadi pada aktuator dan sensor menggunakan komponen *attenuator* dengan perangkat lunak *labVIEW* 2013.

4.3.1 Uji Kesalahan Bias Pada Aktuator dan Sensor Secara Simulasi

a. Hasil respon uji kesalahan bias pada aktuator sebesar 16% dan pada sensor sebesar 8% secara simulasi dapat dilihat pada gambar

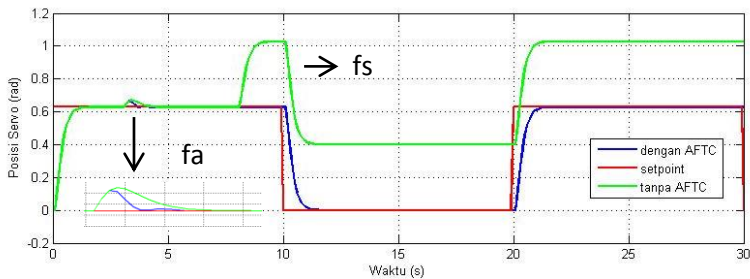


Gambar 4.5. Respon sistem AFTC dengan kesalahan bias 16% pada aktuator dan 8% pada sensor secara simulasi

Tabel 4.1. Performansi sistem dengan kesalahan bias 16% pada aktuator dan 8% pada sensor secara simulasi

Parameter	Kesalahan Aktuator		Kesalahan Sensor	
	Dengan AFTC	Tanpa AFTC	Dengan AFTC	Tanpa AFTC
<i>Overshoot</i>	1,27%	1,6%	0%	0 %
<i>Error Steady State</i>	0%	0%	0%	8 %
<i>Settling Time</i>	4,8 s	5 s	2,2 s	-

b. Hasil respon uji kesalahan bias pada aktuator sebesar 63,7% dan pada sensor sebesar 63,7% secara simulasi dapat dilihat pada gambar



Gambar 4.6. Respon sistem AFTC dengan kesalahan bias 63,7% pada aktuator dan 63,7% pada sensor secara simulasi

Tabel 4 2. Performansi sistem dengan kesalahan bias 63,7% pada aktuator dan 63,7% pada sensor secara simulasi

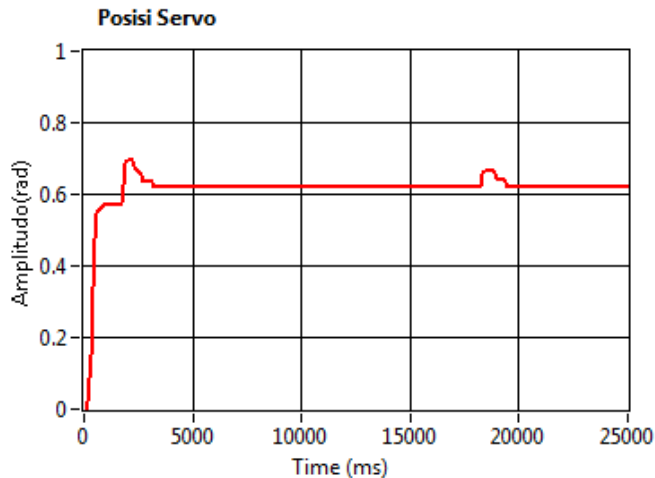
Parameter	Kesalahan Aktuator		Kesalahan Sensor	
	Dengan AFTC	Tanpa AFTC	Dengan AFTC	Tanpa AFTC
<i>Overshoot</i>	5,9%	6,7%	0%	0%
<i>Error Steady State</i>	0%	0%	0%	63,7 %
<i>Settling Time</i>	5 s	5,6 s	2,2 s	-

Pada gambar 4.5 dan 4.6, respon sistem dengan AFTC yang ditunjukkan oleh grafik berwarna biru dan sistem tanpa AFTC yang ditunjukkan grafik berwarna hijau,serta *setpoint* yang ditunjukkan grafik berwarna merah dengan nilai *setpoint* sebesar 0,628 radian. Ketika kesalahan pada aktuator terjadi di detik ke 3, sistem dengan AFTC dan tanpa AFTC mengalami *overshoot* dan sistem dapat kembali mencapai nilai *setpoint*. Sistem dengan AFTC memiliki nilai *overshoot* dan *settling time* yang lebih baik dibanding sistem tanpa AFTC. Ketika kesalahan pada sensor terjadi pada detik ke 8, sistem dengan AFTC tidak mengalami

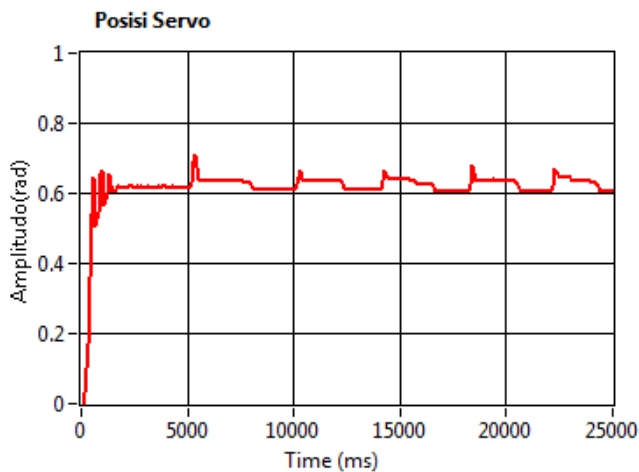
error steady state tetapi pada sistem tanpa AFTC, sistem mengalami *error steady state* dan tidak dapat kembali ke nilai *setpoint*. Hal tersebut terjadi karena sistem dengan AFTC bisa menolerir kesalahan yang ada pada sensor dan aktuator sehingga respon sistem menjadi lebih baik.

4.3.2 Uji Kesalahan Bias Pada Aktuator dan Sensor Secara *Real Time*

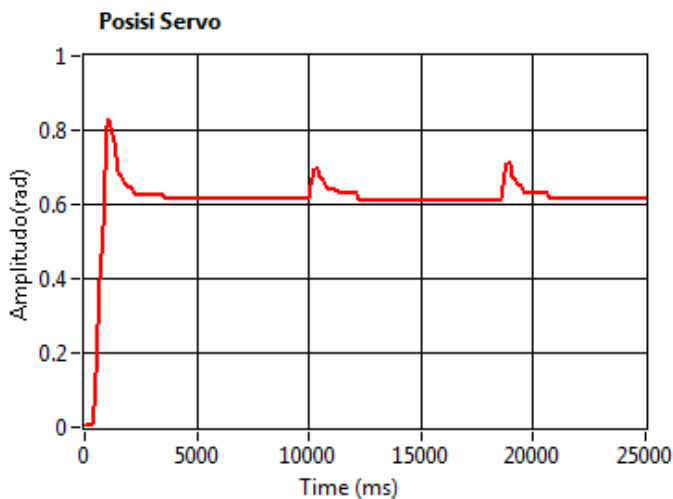
- a. Hasil respon uji kesalahan bias pada pada aktuator sebesar 16% dan 63,7% secara *real time* dapat dilihat pada gambar



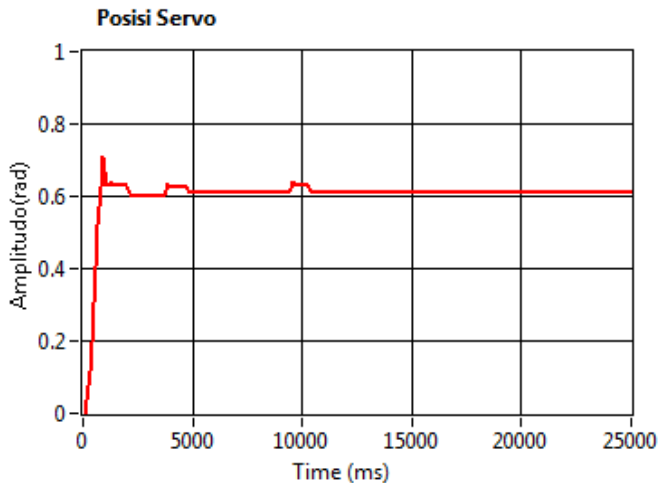
Gambar 4.7. Respon sistem dengan kesalahan bias pada aktuator sebesar 16% secara *real time*



Gambar 4.8. Respon sistem hasil AFTC dengan kesalahan bias pada aktuator sebesar 16% secara *real time*



Gambar 4.9. Respon sistem dengan kesalahan bias pada aktuator sebesar 63,7% secara *real time*



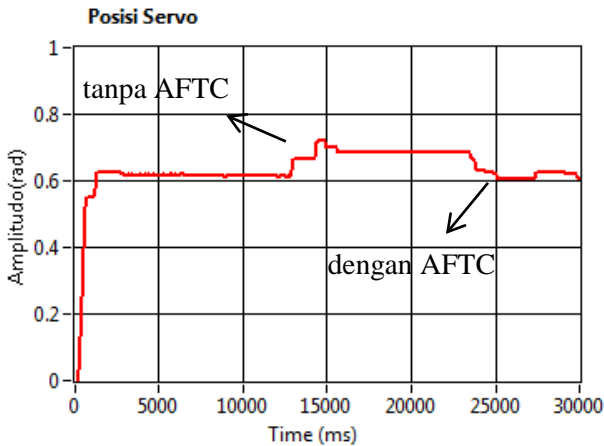
Gambar 4.10. Respon sistem hasil AFTC dengan kesalahan bias pada aktuator sebesar 63,7% secara *real time*

Tabel 4.3. Performansi sistem dengan kesalahan bias pada aktuator 16% dan 63,7% secara *real time*

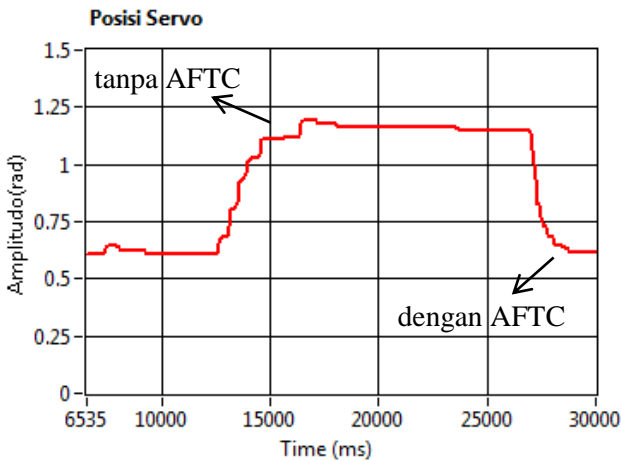
Parameter	Kesalahan 16%		Kesalahan 63,7%	
	Dengan AFTC	Tanpa AFTC	Dengan AFTC	Tanpa AFTC
<i>Overshoot</i>	1,59%	12,7%	11,15%	28,6 %
<i>Settling Time</i>	2,8 s	3,7 s	3 s	4,3 s

Gambar 4.7 dan gambar 4.9 menunjukkan sistem tanpa AFTC sedangkan gambar 4.8 dan gambar 4.10 menunjukkan sistem hasil AFTC dengan nilai *setpoint* sebesar 0,628 radian. Ketika kesalahan bias pada aktuator terjadi, sistem tanpa AFTC memiliki nilai *overshoot* sebesar 12,7% dan 28,6%. Sedangkan sistem dengan AFTC memiliki nilai *overshoot* lebih kecil sebesar 1,59% dan 11,15%. Selain itu *settling time* sistem dengan AFTC lebih baik dari pada sistem tanpa AFTC. Hal tersebut menunjukkan bahwa sistem dengan AFTC bisa menolerir kesalahan yang ada pada aktuator sehingga respon sistem menjadi lebih baik.

- b. Hasil respon uji kesalahan bias pada sensor sebesar 8% dan 63,7% secara *real time* dapat dilihat pada gambar



Gambar 4.11. Respon sistem dengan kesalahan bias pada sensor sebesar 8% secara *real time*



Gambar 4.12. Respon sistem dengan kesalahan bias pada sensor sebesar 63,7% secara *real time*

Tabel 4.4. Performansi sistem dengan kesalahan bias pada sensor 8% dan 63,7% secara *real time*

Parameter	Kesalahan 8%		Kesalahan 63,7%	
	Dengan AFTC	Tanpa AFTC	Dengan AFTC	Tanpa AFTC
<i>Settling Time</i>	2,5 s	-	2,8 s	-
<i>Error</i>	-	11%	-	91%
<i>SteadyState</i>				

Pada gambar 4.11 dan 4.12 diketahui respon sistem dengan AFTC dan tanpa AFTC dengan nilai *setpoint* sebesar 0,628 radian. Ketika kesalahan bias pada sensor terjadi, sistem dengan AFTC dapat memperbaiki kesalahan sehingga dapat mencapai nilai *setpoint*, tetapi pada sistem tanpa AFTC, sistem tidak dapat kembali ke nilai *setpoint* yang diinginkan dan mengalami *error steady state* sebesar 11% dan 91%. Selain itu *settling time* sistem dengan AFTC lebih baik dari pada sistem tanpa AFTC. Hal ini terjadi karena sistem dengan AFTC bisa menolerir kesalahan yang ada pada sensor sehingga respon sistem menjadi lebih baik. Hasil respon sistem tanpa AFTC dan dengan AFTC untuk nilai kesalahan yang lain dapat dilihat pada lampiran.

“Halaman ini memang dikosongkan”

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis data yang telah dilakukan, didapatkan kesimpulan dari tugas akhir mengenai perancangan sistem *Active Fault Tolerant Control (AFTC)* dengan kontrol *cascade* pada pengendalian posisi sistem servo modular MS150 DC adalah sebagai berikut :

- Telah dilakukan perancangan algoritma AFTC berupa perancangan *observer* dan *reconfigurable control* dan kesalahan aktuator dan sensor secara simulasi dan secara *real* pada pengendalian posisi sistem servo modular MS150 DC sehingga sistem mampu memperbaiki kesalahan bias pada aktuator dan sensor dengan sistem dapat kembali mencapai nilai *setpoint* dibandingkan dengan tanpa AFTC.
- Sistem tanpa AFTC pada kesalahan sensor secara simulasi mengalami *error steady state* sebesar 8% dan 63,7% sedangkan sistem dengan AFTC tidak mengalami *error steady state* . Sistem tanpa AFTC pada kesalahan aktuator secara simulasi mengalami *overshoot* sebesar 1,6% dan 6,7%, sedangkan sistem dengan AFTC mengalami *overshoot* lebih kecil sebesar 1,27% dan 5,9%.
- Sistem tanpa AFTC pada kesalahan sensor secara *real time* mengalami *error steady state* sebesar 11% dan 91% sedangkan sistem dengan AFTC tidak mengalami *error steady state* . Sistem tanpa AFTC pada kesalahan aktuator secara *real time* mengalami *overshoot* sebesar 12,7% dan 28,6%, sedangkan sistem dengan AFTC mengalami *overshoot* lebih kecil sebesar 1,59% dan 11,15%.

5.2 Saran

Dalam proses tugas akhir ini terdapat beberapa saran yang dapat digunakan untuk penelitian selanjutnya, yaitu adanya variasi beban dalam melakukan eksperimen pengendalian posisi servo modular MS150 DC.

“Halaman ini memang dikosongkan”

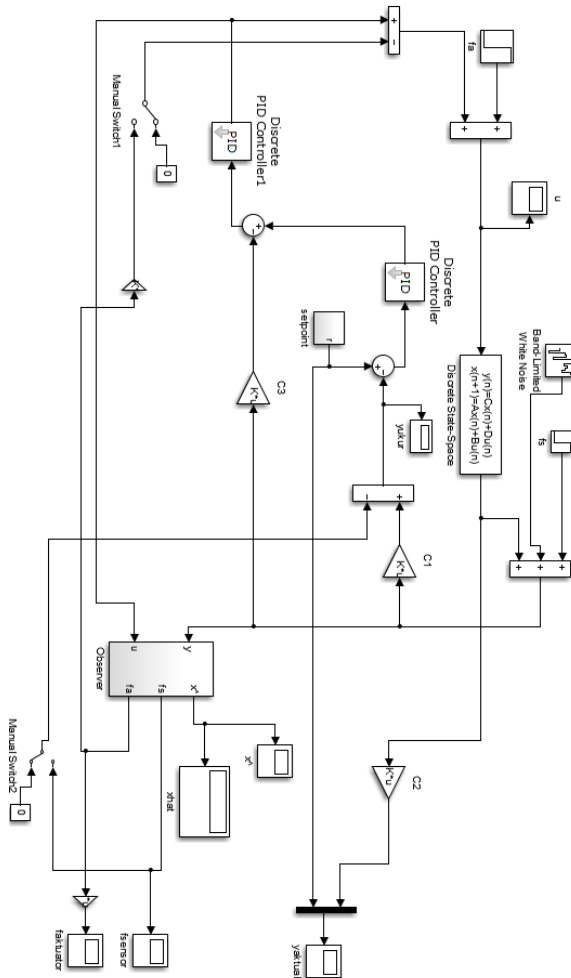
DAFTAR PUSTAKA

- Akbar, I. T. (2017). *Perancangan AFTC pada Sistem MS150DC*. Surabaya: ITS.
- Boom, T. V. (2006). *Discrete Time System Analysis*. Delft: Delft University of Technology.
- Hambley, A. R. (2014). *Electrical Engineering Principles & Applications (6th Edition)*. New Jersey: Pearson Education.
- Indriawati, K. (2015). Robust Fuzzy Observer-Based Fault Tolerant Tracking Control for Nonlinear Systems with Simultaneous Actuator and Sensor Faults: Application to a DC Series Motor Speed Drive. *International Review of Automatic Control (I.R.E.A.CO.)*, Vol. 8, N. 6, 375-385.
- Indriawati, K., Agustinah, T., & Jazidie, A. (2013). Reconfigurable Fault-Tolerant Control of Linear System with Actuator and Sensor Faults. *IEEE*, 22-27.
- Indriawati, K., Agustinah, T., & Jazidie, A. (2015). Robust Fuzzy Observer-Based Fault Tolerant Tracking Control for Nonlinear Systems with Stimultaneous Actuator and Sensor Faults: Application to a DC Series Motor Speed Drive. *Praise Worthy Prize, VIII*(6), 375-385.
- Instruments. (2013). *Control & Instrumentation*. East Sussex: Feedback Instruments Ltd.
- Instruments, N. (2004). *labVIEW System Identification Toolkit User Manual*. USA: National Instruments.
- M. Chen, B. (2016). *Digital Control System*. Singapore: The National University of Singapore.
- Mahmoud, M. (2003). *Active Fault Tolerant Control System*. Berlin: Springer.

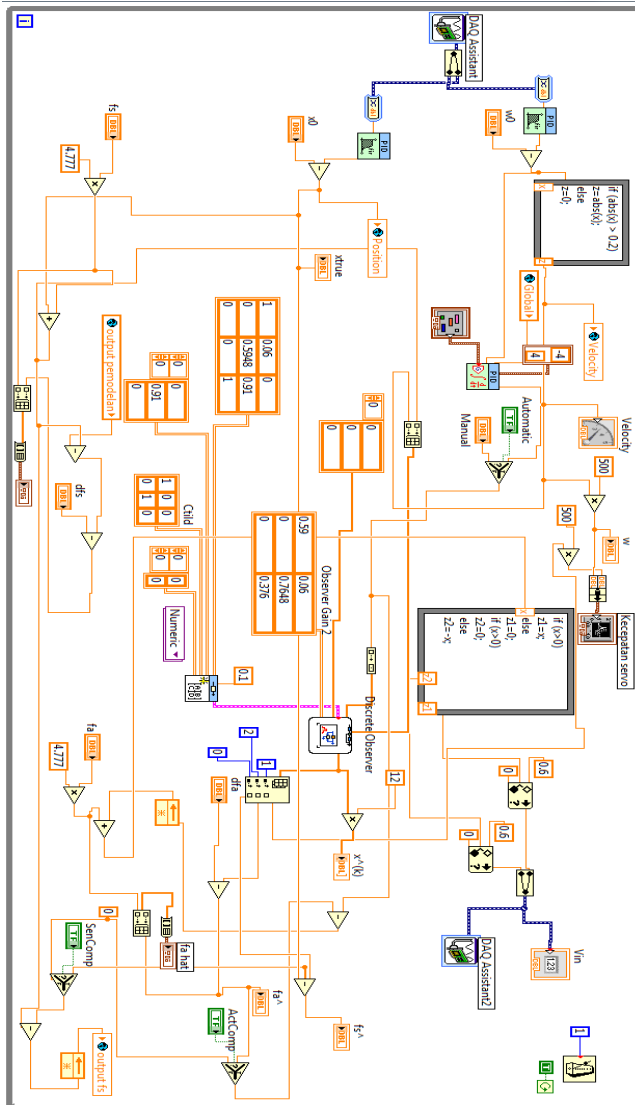
- Namazov, M. (2010). DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods. *Turkish Journal of Fuzzy Systems*, Vol.1, No.1, pp. 36-54.
- Ogata, K. (2004). *System Dynamics*. New Jersey: Prentice-Hall.
- Ogata, K. (2010). *Modern Control Engineering*. United States of America: Pearson Prentice Hall.
- Sumanto. (1994). *Mesin Arus Searah*. Jogjakarta: ANDI OFFSET.
- Tushar Jain, J. Y. (2017). *Active Fault Tolerant Control System*. Cham: Springer.
- Visioli, A. (2006). *Practical PID Control*. Brescia: Springer.
- Wildi, T. (2002). *Electrical Machines, Drives, and Power Systems*. New Jersey: Prentice Hall.
- Zhang, Y., & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Science Direct*, 229-252.

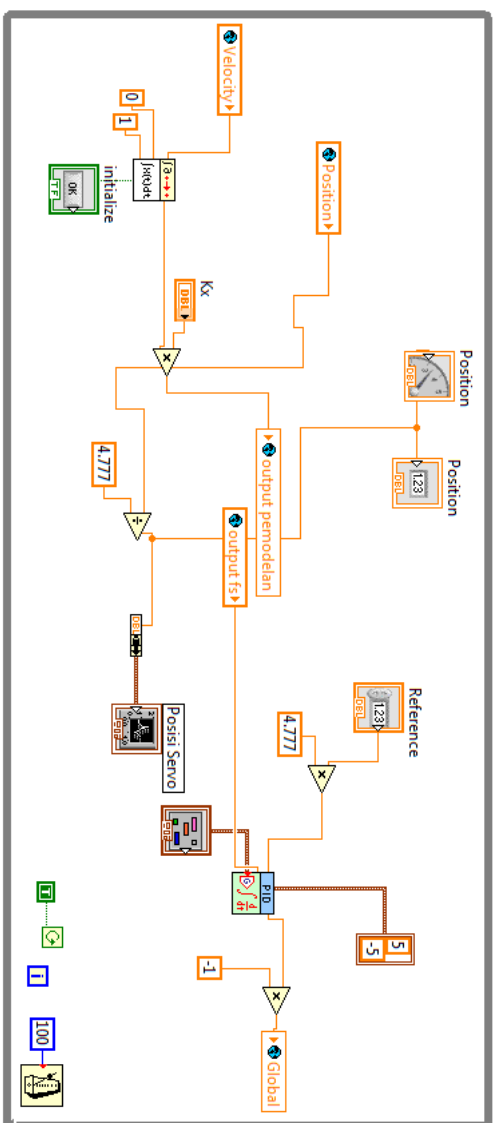
LAMPIRAN

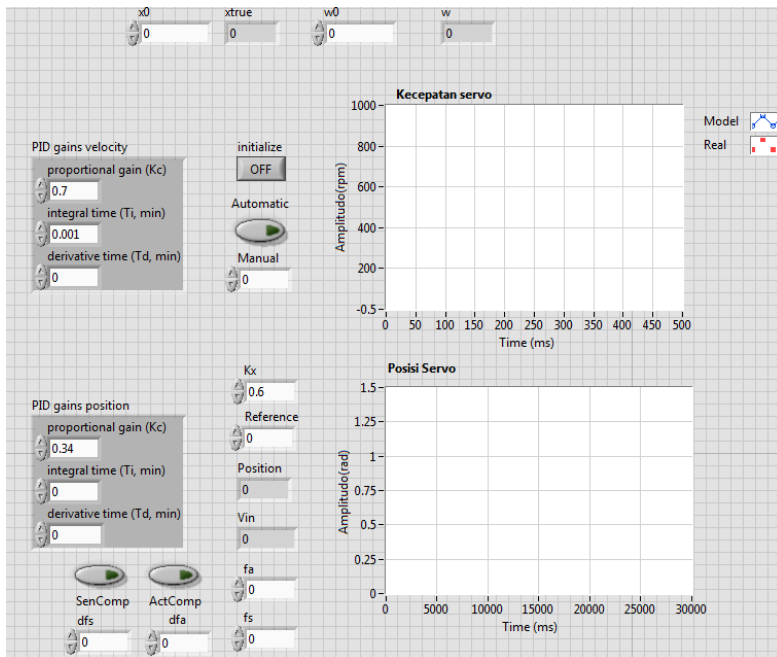
A. Simulasi Penerapan AFTC pada program *MATLAB* R2013A



B. Penerapan AFTC secara *real time* pada program *labVIEW 2013*

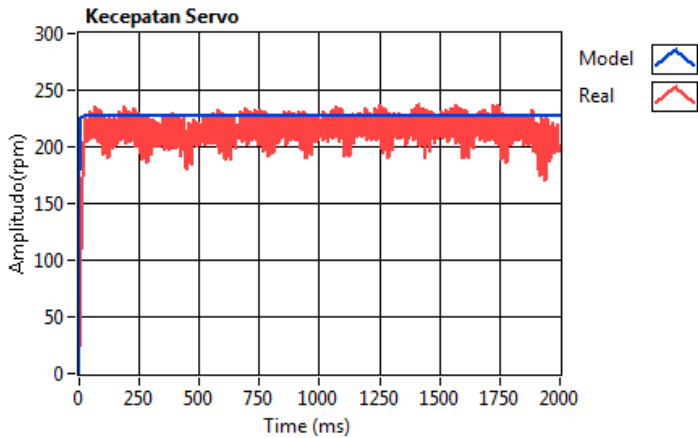




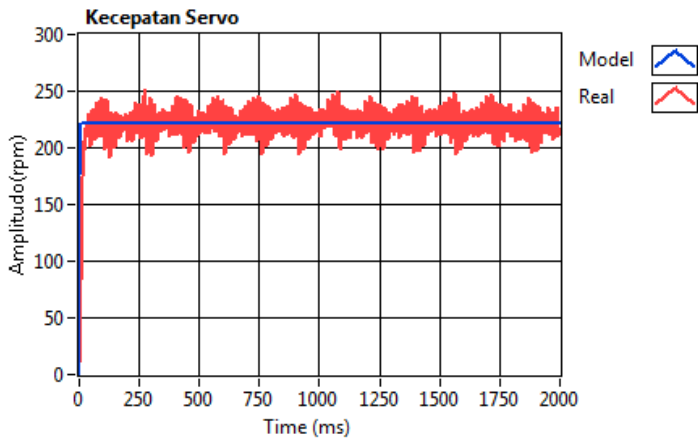


C. Grafik validasi model kecepatan

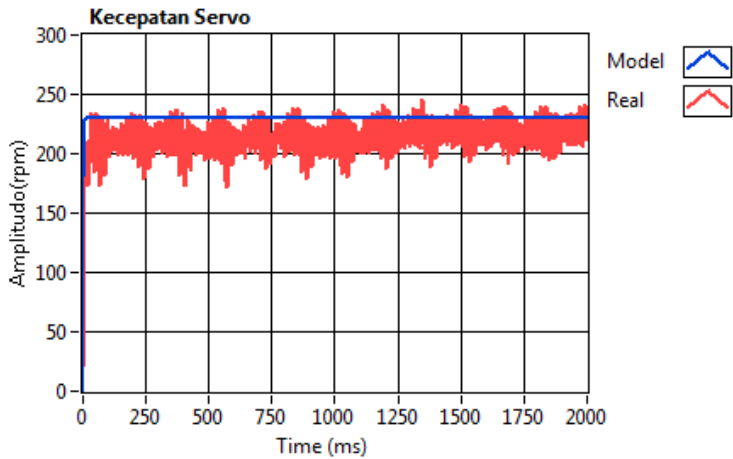
- $$\omega(z) = \frac{0.92}{z - 0.5948} u(z)$$



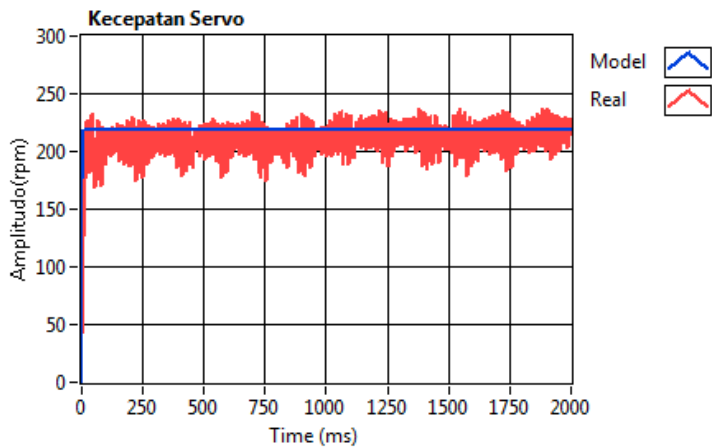
- $$\omega(z) = \frac{0.9}{z - 0.5948} u(z)$$



- $$\omega(z) = \frac{0.91}{z - 0.6048} u(z)$$

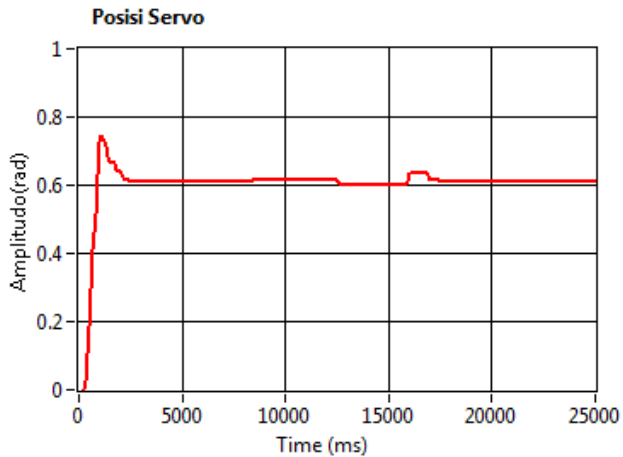


- $$\omega(z) = \frac{0.91}{z - 0.5848} u(z)$$

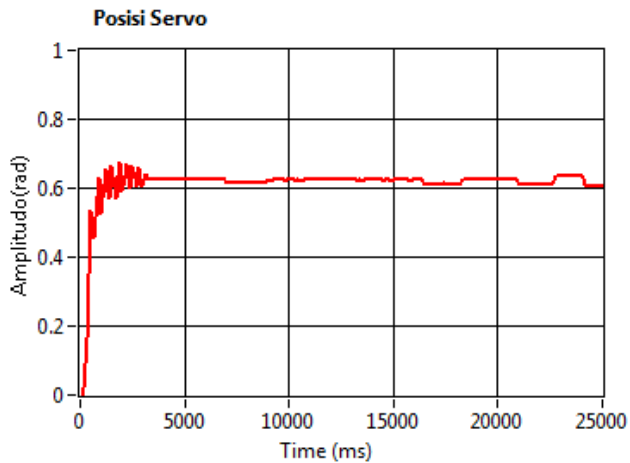


D. Grafik sistem tanpa AFTC dan dengan AFTC

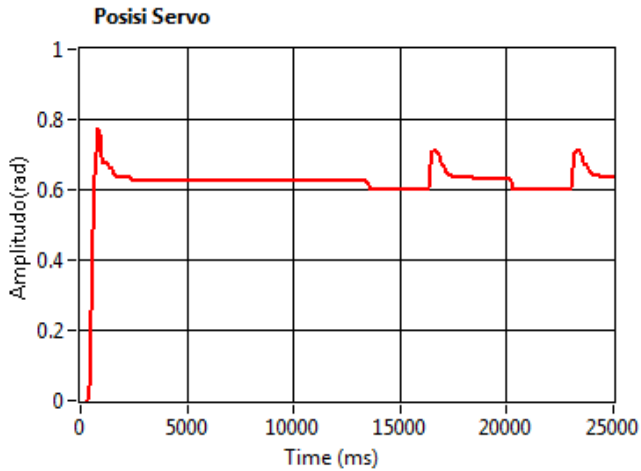
- Respon sistem dengan kesalahan bias pada aktuator sebesar 31,8% secara *real time*



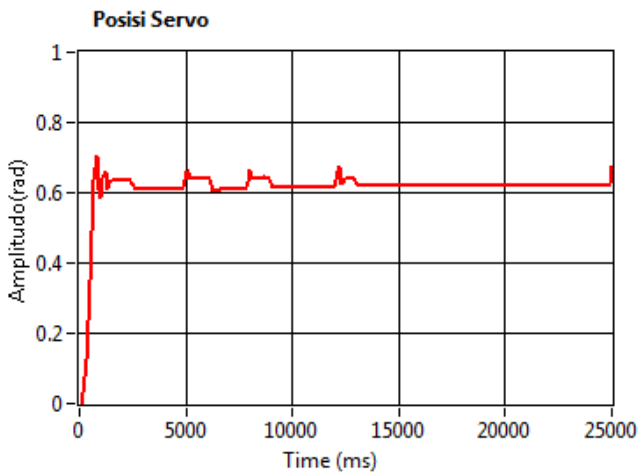
- Respon sistem hasil AFTC dengan kesalahan bias pada aktuator sebesar 31,8% secara *real time*



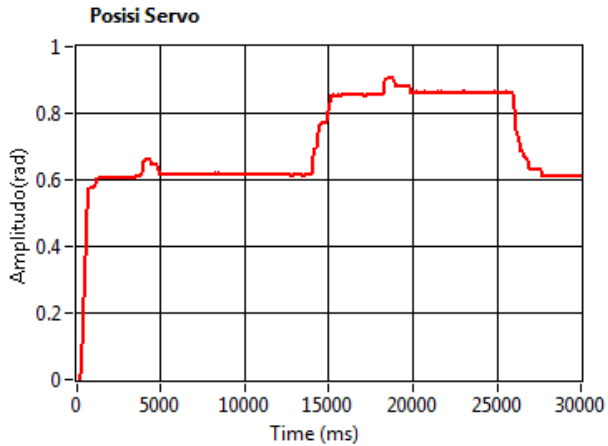
- Respon sistem dengan kesalahan bias pada aktuator sebesar 47,7% secara *real time*



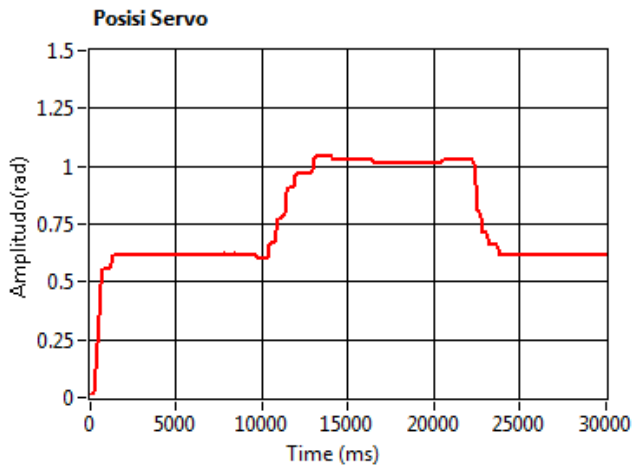
- Respon sistem hasil AFTC dengan kesalahan bias pada aktuator sebesar 47,7% secara *real time*



- Respon sistem dengan kesalahan bias pada sensor sebesar 31,8% secara *real time*



- Respon sistem dengan kesalahan bias pada sensor sebesar 47,7% secara *real time*



E. Tabel konversi sudut posisi servo

Posisi servo (rad)	Tegangan (volt)
0,209	1
0,419	2
0,628	3
0,838	4
1,047	5

BIODATA PENULIS



Nama penulis Tulus Indra Hermawan. Penulis dilahirkan di Mojokerto, tanggal 7 Agustus 1994 dari ayah adalah Heri Widodo dan ibu adalah Ainin Kholianah. Saat ini penulis tinggal di Dusun Badung, Desa Kedunglengkong, Kecamatan Dlanggu, Kabupaten Mojokerto, Provinsi Jawa Timur. Penulis telah menyelesaikan pendidikan tingkat dasar di SDN Kedunglengkong II Mojokerto pada tahun 2007, pendidikan menengah pertama di SMPN 1 Puri Mojokerto

pada tahun 2010, pendidikan menengah atas di SMAN 1 Puri Mojokerto pada tahun 2013, dan sedang menempuh pendidikan S1 Teknik Fisika FTI di Institut Teknologi Sepuluh Nopember Surabaya hingga sekarang.

Pada bulan Maret 2018 penulis telah menyelesaikan Tugas Akhir dengan judul **Perancangan Sistem Active Fault Tolerant Control (AFTC) untuk Pengendalian Posisi dengan Kontrol Cascade pada Sistem Servo Modular MS150 DC**. Bagi pembaca yang memiliki saran, kritik, atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini maka dapat menghubungi penulis melalui *email*: tulus.acm@gmail.com.